# IUGN 17

CONTENTS



DATA PROTECTION ACT 1985:
Details of the members of the IUGN are held on computer file. Each member may view, alter or destroy any data held on him/her within that file. To obtain a copy of your file please send a stamped addressed envelope to the Editor. Your subsequent wishes regarding that data will be honoured without question.

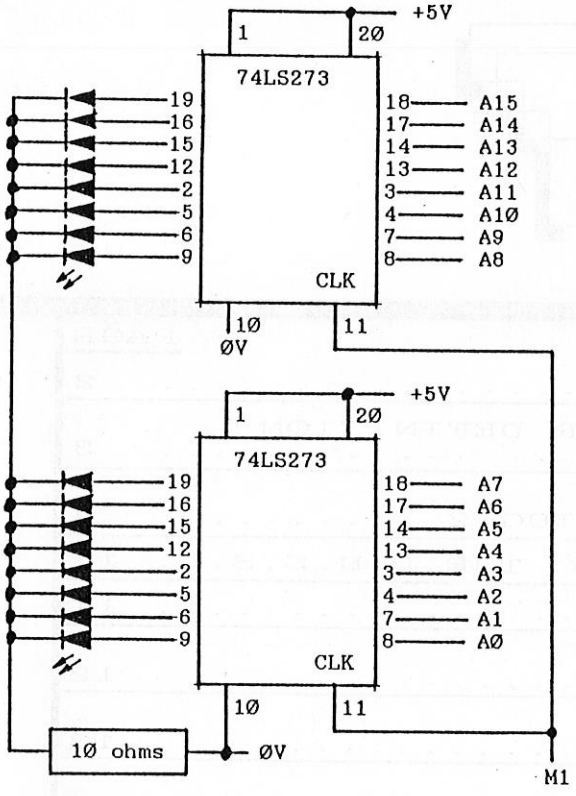
NOTES

==========

CI-AM

CURRENT INSTRUCTION ADDRESS MONITOR

This circuit can be added to the patch area on the MZB-3 CPU card to indicate the current instruction address. It is updated at each op-code fetch. You will need to drill some extra holes next to the patch area to carry the larger sockets, but it is no problem to do.




Mount the 16 LED's on the front panel to indicate the current instruction address as :-

| | |
|---|---|
| O O O O | A15 A14 A13 A12 |
| O O O O | A11 A1Ø A9 A8 |
| O O O O | A7 A6 A5 A4 |
| O O O O | A3 A2 A1 AØ |

Greenbank do a miniture green LED which is perfect for the job.

==========

8Ø COL VDU

I have heard that the new 8Ø column VDU design is now past the prototyping phase and is working. The next job is to lay it out on a card for release. It sounds very powerful and will probally require two cards to fit it all on. It is all still something of a secret at the moment, but I have heard that high resolution graphics may be built in and the device will be port mapped to free that last page of memory to the system. I will report any news as fast as I get it. I have already got my order in for the first issue of the card.

==========

COCONUTS

Three natives spent all day collecting a pile of coconuts. They agree to sleep and share out the pile in the morning. During the night one of the natives woke-up and decided to save time by dividing the pile into three heaps. He found that he had an odd coconut left over and so he threw it away into the trees. Taking up his pile he carefully hid it and then mixed the remaining coconuts together before going back to sleep. A little while later another native woke-up and he decided that he could save time in the morning. He divided the heap of coconuts into three piles and throwing away the odd one that was left, he hid his third, finally mixing the remainder into a heap and going back to bed. Later still the third native woke-up and he divided the heap into three piles. He also threw away the odd coconut that was left over. He hid his third and after mixing the remaining coconuts into a nice heap he went back to sleep. In the morning the three natives divided the heap into three piles, they discarded the odd coconut that was left over and took their coconuts away with them.

How many coconuts were in the original pile?.

==========

RPM

RAM PAGE MONITOR

If you would like an indication of the current 4K page being accessed. A simple disply can be made on the LKP-1 card using 16 LEDS and the existing but redundant 74LS154 page decoder. Just connect one LED to each output (pad P4 and P3) and arrange the LEDS on the front panel as :-

| | |
|---|---|
| O O O O | PF PE PD PC |
| O O O O | PB PA P9 P8 |
| O O O O | P7 P6 P5 P4 |
| O O O O | P3 P2 P1 PØ |

In this case the outputs are active low, so the other end of the LED's should go to +5V via a resistor of around 33Ø ohms. I suggest you try one LED on a leg to establish the required resistor value to get brightness you like. Feed all 16 LED's from the chosen resistor, a trim pot will give brightness control. Super stars could decode the display to a 'Hex display chip' so as to show the page as a number Ø thru F.

==========

POWER DISTRIBUTION CARD

I recently fitted a *PSD card to my Interak. What a delight it is to remove all the untidy hodge-podge of power wiring. The PSD comes with sufficient plugging to seperate all of the various supplies for the machine so that everything is unpluggable from everything else. Also full earthing can be achived throughout the entire machine. Very safe and long lasting. I recommend that you fit one A.S.A.P.

==========

BACK ISSUES

D.Parkins, Greenbank Electronics, 46Ø New Chester road, Rock Ferry, Birkenhead, Merseyside, L42 2AE. Phone Ø51-645-3391

BOOK LIBRARIAN

R.E.Bowyer, 45 Ford drive, Yarnfield, Stone, Staffs.

DISK LIBRARIAN

C.V.Bridgstock, 32 Wimborne ave, Thingwall,Wirral, Merseyside, L61 7UL. Phone Ø51-648-3ØØØ.

EDITOR (IUGN)

R.Eldridge, 28 Wycherley Close, Blackheath, London, SE3 7QH.

MEMBERSHIP SECRETARY

Tom Evans, 129 Cranborne Waye, Hayes, Middlesex, UB4 OHR.

PROGRAMMABLE CHARACTER DEFINITION CARD
Constructional details
By F.R. Johnson

In responce to several requests for further information regarding my "plug-in" Programmable Character Definition card, I have drawn some diagrams and notes to show how I constructed it.

I first built the circuit on a breadboard card, but found that despite pull-up resistors and copious decoupling I could not get a steady display. The problem appeared to be interference being picked up by the ribbon cable, since I could cure most of my troubles by having an extremely short cable. Following this to its natural conclusion led to the idea of a daughter board plugged into the VDU card.

In order to make the necessary connections between the boards I used "Vero-pins" soldered to the daughter board. 31 of these are plugged into corresponding SIL sockets, which are soldered into convenient holes on the VDU card bus, (thanks are due here to the forethought of the VDU designer!). The remaining, slightly shortened pins are pushed into a 24 pin DIL socket which is in turn plugged into the "Application Characters" socket. This double arrangement prevents any damage to the socket on the VDU card and makes lining up of the pins easier.

My daughter board was constructed from "square pad Vero-board" and "Vero-wire", but other constructors may wish to produce their own PCB's or use a breadboard card. The latter idea is a bit wasteful, however, since you cannot make use of the edge connector. The two boards are arranged to be exactly 1" apart so that with suitable spacing washers, an additional 5E card front can be fitted to the daughter board or the existing 5E front changed for a 10E version.

The diagrams show the relative positions of the two boards, and apart from the position of the connecting pins, the layout depends on the type of board being used.

I hope this enables other members to get a programmable character card up and running.

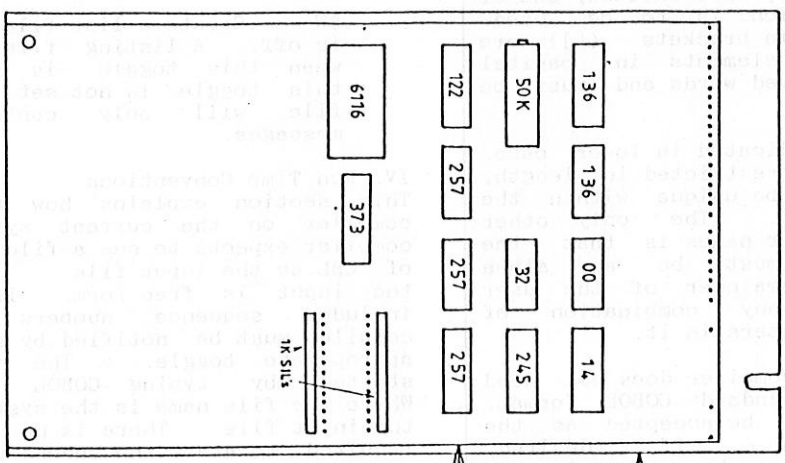



Fig 1

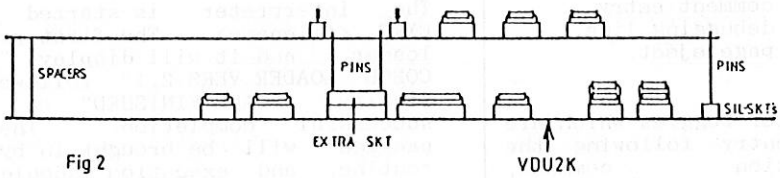



Fig 2

COBOL.DOC
A document file from IUG-2

NPS MICRO-COBOL Ver. 2.1
User's Guide

I. Organization

The compiler is designed to run on an 8080 system in an interactive mode through the use of a teletype or console. It requires at least 24k of main memory and a mass storage device for reading and writing. The compiler is composed of two parts, each of which reads a portion of the input file. Part One reads the input program to the end of the Data Division and builds the symbol table. At the end of the Data Division, Part One is overlayed by Part Two which uses the symbol table to produce the code. The output code is written as it is produced to minimize the use of internal storage.

The EXEC Program builds the core image for the intermediate code and performs such functions as backstuffing addresses and offsetting address in subroutines. EXEC then copies the interpreter (CINTERP.COM) into memory and transfers control to it. The interpreter is controlled by a large case statement that decodes the instructions and performs the required actions.

II. MICRO-COBOL Elements

This section contains a description of each element in the language and shows simple examples of their use. The following conventions are used in explaining the formats: Elements enclosed in broken braces (<>) are themselves complete entites and are described elsewhere in the manual. Elements enclosed in braces({}) are choices, one of the elements which is to be used. Elements enclosed in brackets ([]) are optional. All elements in capital letters are reserved words and must be spelled exactly.

User names are indicated in lower case. These names are unrestricted in length, however they must be unique within the first 15 characters. The only other restriction on user names is that the first character must be an alpha character. The remainder of the user name can have any combination of representable characters in it.

The input to the compiler does not need to conform to standard COBOL format. Free form input will be accepted as the default condition. If desired, sequence numbers can be entered in the first six positions of each line. However, a toggle needs to be set to cause the compiler to ignore the sequence numbers.

The first character position on any line is used to indicate the following:

* - indicates a comment entry.
: - indicates a debugging line.
/ - indicates a page eject.

III. Compiler Toggles

There are six compiler toggles which are controlled by an entry following the compiler activation command, COBOL<filename>. The format of the entry consists of following <filename> by one space and then entering a "$" followed immediately by the desired toggles. There must be only one space after <filename> and no spaces between the "$" and the toggles. The following is an example of a typical entry:

COBOL EXAMPLE $S

This entry would cause the compiler to ignore the first six characters (used for sequence numbers) at the beginning of each input line. In each case the toggle reverses the default value.

$C - No intermediate code. Default is off. Setting this toggle speeds initial compilation for syntax checking. When this toggle is set the "CIN" file is empty.

$D - Debugging mode. Default is off. This toggle sets the debugging mode, which means all debugging lines (those with a ':' in column one) are compiled. If this toggle is not set in the ENVIRONMENT DIVISION of the source program all debugging lines are treated as comments.

$L - list the input code on the screen as the program is compiled. Default is on. Error messages are displayed at the terminal in any case.

$P - Productions. List productions as they occur. Default is off.

$S - sequence numbers are in the first six positions of each record. Default is off.

$T - Tokens. List tokens from the scanner. Default is off.

$W - Create a list file. Default is off. A listing file is created when this toggle is set. When this toggle is not set the "LST" file will only contain error messages.

IV. Run Time Conventions

This section explains how to run the compiler on the current system. The compiler expects to see a file with a type of CBL as the input file. In general, the input is free form. If the input includes sequence numbers then the compiler must be notified by setting the appropriate toggle. The compiler is started by typing COBOL <filename>. Where the file name is the system name of the input file. There is no interaction required to start the second part of the compiler. The output file will have the same <filename> as the input file, and will be given a file type of CIN. any previous copies of the file will be erased. As with the CIN file, a LST file will be created with the same file name as the input file and any previous LST files with that name will Perased.

The interpreter is started by typing EXEC <filename>. The first program is a loader, and it will display "NPS MICRO-COBOL LOADER VERS 2.1" followed by the display "LOAD FINISHED" to indicates successful completion. The run-time package will be brought in by the EXEC routine, and execution should continue without interruption. Successful transfer of control to the interpreter will be indicated by the display "NPS MICRO-COBOL INTERPRETER VERS 2.1." Completion of program execution will be indicated by the display "X EXECUTION

ERROR(S)," where "X" is the number of errors which occurred during execution.

V. File Interactions with CP/M
The file structure that is expected by the program imposes some restrictions on the system. References 4 and 5 contain detailed information on the facilities of CP/M, and should be consulted for details. The information that has been included in this section is intended to explain where limitations exist and how the program interacts with the system.

All files in CP/M are on a random access device, and there is no way for the system to distinguish sequential files from files created in a random mode. This means that the various types of reads and writes are all valid to any file that has fixed length records. The restrictions of the ASSIGN statement prevent a file from being open for both random and sequential actions during one program.

Each logical record is terminated by a carriage return and a line feed. In the case of variable length records, this is the only end mark that exists. This convention was adopted to allow the various programs which are used in CP/M to work with the files. Files created by the editor, for example, will generally be variable length files. This convention removes the capability of reading variable length files in a random mode.

All of the physical records are 128 bytes in length, and the program supplies buffer space for these records in addition to the logical records. Logical records may be of any desired length.

---

IDENTIFICATION DIVISION

Element: IDENTIFICATION DIVISION Format

```
Format:  IDENTIFICATION DIVISION.
         PROGRAM-ID. <comment>.
         [AUTHOR. <comment>.]
         [DATE-WRITTEN. <comment>.]
         [SECURITY. <comment>.]
```

Description:
This division provides information for program identification for the reader. The order of the lines is fixed.

Examples:

```
     IDENTIFICATION DIVISION.
     PROGRAM-ID. SAMPLE.
     AUTHOR. HAL R POWELL.
```

---

ENVIRONMENT DIVISION

Element: ENVIRONMENT DIVISION Format

Format:

```
[ ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER. <comment> [DEBUG MODE].
OBJECT-COMPUTER. <comment>.
[INPUT-OUTPUT SECTION.
FILE-CONTROL.
   <file-control-entry> . . .[I-O-CONTROL
 .SAME file-name-1 file-name-2[file-
 -name-3][file-name-4][file-name-5] ] ] ]
```

Description:
This division determines the external nature of a file. In the case of CP/M all of the files used can be accessed either sequentially or randomly except for variable length files which are sequential only. The debugging mode is also set by this section. The DEBUGGING MODE clause is used in conjunction with the ':' to indicate conditional compilation. If this clause is specified, all debugging lines (those with a ':' in column one) are compiled. If this clause is not specified, all debugging lines are treated as comments. In addition the DEBUGGING MODE can be specified by using the compiler toggle 'D.'

---

Element: <file-control-entry>

Format:

```
1.SELECT file-name
   ASSIGN implementor-name
   [ORGANIZATION SEQUENTIAL]
   [ACCESS SEQUENTIAL].

2.SELECT file-name
   ASSIGN implementor-name
   ORGANIZATION RELATIVE
   [ACCESS {SEQUENTIAL [RELATIVE data-name
   ]}].
           {RANDOM RELATIVE data-name
   }

3.SELECT file-name
    ASSIGN implementor-name
    ORGANIZATION INDEXED
    [ACCESS {SEQUENTIAL}].
            {RANDOM    }
```

Description:
The file-control-entry defines the type of file that the program expects to see. There is no difference on the diskette, but the type of reads and writes that are performed will differ. For CP/M the implementor name needs to conform to the normal specifications. Indexed is not implemented.

Examples:

```
     SELECT CARDS
          ASSIGN CARD.FIL.
     SELECT RANDOM-FILE
          ASSIGN A.RAN
          ORGANIZATION RELATIVE
          ACCESS RANDOM RELATIVE RAND-FLAG.
```

---

DATA DIVISION

Element: DATA DIVISION Format

Format:

```
DATA DIVISION.
[FILE SECTION.
[FD file-name
    [BLOCK integer-1 RECORDS]
    [RECORD [integer-1 TO] integer-3]
    LABEL RECORDS {STANDARD}]
                  {OMITTED}
    [VALUE  OF implementor-name-1 literal-
    1[implementor-name-2 . . . .literal-2
    ] . . ][<record-description-entry>] ]
    [WORKING-STORAGE SECTION.
    [<record-description-entry>] . . . ]
    [LINKAGE SECTION.
    [<record-description-entry>] . . . ]

                    <comment>
```

Description:
This is the section that describes how the data is structured. There are no major differences from standard COBOL except for the following: 1. Label records make no sense on the diskette so no entry is required. 2. The VALUE OF clause likewise has no meaning for CP/M.

If a record is given two lengths as in RECORD 12 to 128, the file is taken to be variable length and can only be accessed in the sequential mode. See the section on files for more information.

---

Element: <comment>

Format: any string of characters

Description:
A comment is a string of characters. It may include anything other than a period followed by a blank or a reserved word, either of which terminate the string. Comments may be empty if desired, but the terminator is still required by the program.

Examples:
```
     this is a comment
     anotheroneallruntogether
     8Ø8Øb 16K
```

Note page 1: * in column 1 to cause compiler to ignore sequence numbers.

---

Element: <data-description-entry> Format

Format:
```
level-number {data-name}
                {FILLER   }
        [REDEFINES data-name]
        [PIC character-string]
        [USAGE {COMP    }]
               {COMP-3}
               {COMPUTATIONAL}
               {DISPLAY}
        [SIGN {LEADING} [SEPARATE]]
              {TRAILING}
        [OCCURS integer]
        [SYNC [LEFT]]
              [RIGHT]
        [VALUE literal].
```

Description:
This statement describes the specific attributes of the data. Since the 8Ø8Ø is a byte machine, there was no meaning to the SYNC clause, and thus it has not been implemented, however existing programs that are transferred to MICRO-
COBOL and use this feature will compile and execute successfully. All numeric data are maintained in DISPLAY format or packed BCD if the COMP-3 option is used.

Examples:
```
Ø1 CARD-RECORD.
  Ø2 PART PIC X(5).
  Ø2 NEXT-PART PIC 99V99 USAGE DISPLAY.
  Ø2 FILLER.
     Ø3 NUMB PIC S9(3)V9 SIGN LEADING
        SEPARATE
     Ø3 LONG-NUMB 9(15).
     Ø3 STRING REDEFINES LONG-NUMB PIC
        X(15).
  Ø2 ARRAY PIC 99 OCCURS 1ØØ.
```

---

PROCEDURE DIVISION

Element: PROCEDURE DIVISION Format

Format:
```
1.
PROCEDURE DIVISION [USING name1 [name2] .
 . . . . . [name5]].
                    section-name SECTION.
[paragraph-name. <sentence> [<sentence>
 . . . ] . . . ] . . .
```

```
2.
PROCEDURE DIVISION [USING name1 [name2]
 . . . . . [name5]].
    paragraph-name. <sentence> [ . . .
 <sentence>. . . ] . . .
```

Description:
As is indicated, if the program is to contain sections, then the first paragraph must be in a section.

---

Element: <sentence>

Format:
```
<imperative-statement>
<conditional-statement>
```

---

Element: <imperative-statement>

Format: The following verbs are always imperative:

| | |
|---|---|
| ACCEPT | 15 |
| CALL | 17 |
| CLOSE | 17 |
| DISPLAY | 18 |
| EXIT | 19 |
| GO | 2Ø |
| MOVE | 22 |
| OPEN | 24 |
| PERFORM | 25 |
| STOP | 27 |

The following *may be* imperatives: arithmetic verbs (ADD (16), DIVIDE (19), MULTIPLY (23), SUBTRACT (28)) *without* the SIZE ERROR statement and DELETE (18), WRITE (29), and REWRITE (26) *without* the INVALID option.

---

Element: <conditional-statements>

Format:

| | |
|---|---|
| IF | 21 |
| READ | 26 |

arithmetic verbs *with* the SIZE ERROR statement and DELETE, WRITE, and REWRITE *with* the INVALID option.

---

Element: ACCEPT

Format: ACCEPT<identifier>

Description:
This statement reads up to 255 characters from the console. The usage of the item must be DISPLAY.

Examples:
```
     ACCEPT IMAGE.
     ACCEPT NUM(9).
```

---

Element: ADD

Format:
```
ADD{identifier-1}[{identifier-2}] . . .
 TO identifier-m
   {literal-1   } {literal-2   }
[ROUNDED] [SIZE ERROR <imperative-
statement>]
```

Description:
This instruction adds either one number to a second with the result being placed in the last location. Multiple adds have not been implemented.

Examples:
```
     ADD 1Ø TO NUM1
     ADD X TO Z ROUNDED.
     ADD 1ØØ TO NUMBER SIZE ERROR GO
     ERROR-LOC
```

---

```
Element: CALL

Format:
 CALL literal [USING name1 [name2] . . .
 . . . [nameN]]

Description:
Control is transferred to the called
procedure with an address of each of the
parameters to be passed. The parameters
map to those in the linkage section of
the called program. The type and
size of the parameters must match
exactly.

Examples:
     CALL 'NC152' USING DN1
     CALL 'PRINT'
     CALL 'ADDLIST' USING VAR1 VAR2 VAR3
-----------------------------------------

Element: CLOSE

Format:  CLOSE file-name

Description:
Files must be closed if they have ben
written. However, the normal
requirement to close an input file
prior to the end of processing does not
exist.

Examples:
     CLOSE FILE1
     CLOSE RANDFILE
-----------------------------------------

Element: DELETE

Format:

DELETE file-name [INVALID <imperative-
statement>]

Description:
This statement requires the file-name of
the item to be deleted. The record is
logically removed by filling it with a
high value character, which is not
displayable to the console or line
printer. The logical record space can be
used again by writing a valid record in
its place.

Examples:
     DELETE FILE-NAME
-----------------------------------------

Element: DISPLAY

Format:
DISPLAY{identifier}[{identifier-1}] . . .
[{identifier-N}]
      {literal   } {literal-1   } . . .
 {literal-N  }

Description:
This displays the contents of an
identifier or displays a literal on the
console. Usage must be DISPLAY. The
maximum length of the display is 80
characters for literal values and 255
characters for identifiers.

Examples:
     DISPLAY MESSAGE-1
     DISPLAY MESSAGE-3 10
     DISPLAY 'THIS MUST BE THE END'
-----------------------------------------

Element: DIVIDE

Format:
DIVIDE{identifier}INTO identifier-1
[ROUNDED]
      {literal   }
[SIZE ERROR <imperative-statement>]

Description:
The result of the division is stored in
identifier-
1; any remainder is lost.

Examples:
     DIVIDE NUMB INTO STORE
     DIVIDE 25 INTO RESULT
-----------------------------------------

Element: EXIT

Format:  EXIT [PROGRAM]

Description:
The EXIT command causes no action by the
interpreter but allows for an empty
paragraph for the construction of a
common return point. The optional
PROGRAM terminates a subroutine and
returns to the calling program. It's
use in the main program causes no action
to be taken.

Examples:
     EXIT PROGRAM
     EXIT
-----------------------------------------

Element: GO

Format:
1. GO procedure-name
2. GO procedure-1 [procedure-2] . . .
procedure-20
          DEPENDING identifier

Description:
The GO command causes an unconditional
branch to the routine specified. The
second form causes a forward branch
depending on the value of the contents of
the identifier. The identifier must be a
numeric integer value. There can be no
more than 20 procedure names.

Examples:
GO READ-CARD.
GO READ1 READ2 READ3 DEPENDING READ-INDEX
-----------------------------------------

Element: IF

Format:
IF <condition> {stmt-lst} END-IF
IF <condition> {stmt-lst} ELSE {stmt-lst}
 END-IF
{NEXT SENTENCE}   {NEXT SENTENCE}

Description:
This is an enhanced version of the
standard COBOL IF statement. Nesting of
IF statement is allowed.

Examples:
IF A GREATER B ADD A TO C ELSE GO ERROR-
ONE END-IF.

IF  A NOT NUMERIC NEXT SENTENCE ELSE MOVE
 ZERO TO A END-IF.

IF A L
IF A GREATER B ADD A TO C ELSE GO ERROR-
ONE END-IF.

IF  A NOT NUMERIC NEXT SENTENCE ELSE MOVE
ZERO TO A END-IF.

IF A LESS B
     DISPLAY A
     DISPLAY B END-IF.

IF A GREATER B
     DISPLAY A
     DISPLAY B
ELSE
```

```
     DISPLAY C
     DISPLAY D END-IF.

IF A GREATER B
     IF A GREATER C
        DISPLAY A
     ELSE
        DISPLAY C
     END-IF
ELSE
     IF B GREATER C
        DISPLAY B
     ELSE
        DISPLAY C
     END-IF
END-IF.
```

---

Element: MOVE

Format:
```
MOVE {identifier-1} TO identifier-2
     {literal     }
```

Description:
The standard list of allowable moves applies to this action. As a space saving feature of this implementation, all numeric moves go through the accumulators. This makes numeric moves slower than alpha-numeric moves, and where possible they should be avoided. Any move that involves picture clauses that are exactly the same can be accomplished as an alpha-numeric move if the elements are redefined as alpha-numeric; also all group moves are alpha-numeric.

Examples:
```
     MOVE SPACE TO PRINT-LINE.
     MOVE A(1Ø) TO B(PTR).
```

---

Element: MULTIPLY

Format:
```
MULTIPLY {identifier} BY identifier-2
[ROUNDED]
         {literal    }
[SIZE ERROR <imperative-statement>]
```

Description:
The multiply routine uses a double length register to calculate the result. This allows the result generated to be of maximum precision. The actual value stored will be determined by the amount of storage allocated for the variable. Overflow will occur if the number in the register is larger than the variable. If the precision in the register is greater than the variable, truncation occurs unless the round option is specified.

Examples:
```
MULTIPLY X BY Y.
MULTIPLY A BY B(7) SIZE ERROR GO OVERFLOW.
```

---

Element: OPEN

Format:
```
OPEN {INPUT file-name-1 } [{file-name-2}]
     {OUTPUT file-name-1} [{file-name-2}]
     {I-O file-name-1    } [{file-name-2}]
```

Description:
The three types of OPENS have exactly the same effect on the diskette. However, they do allow for internal checking of the other file actions. For example, a write to a file set open as input will cause a fatal error. Multiple opens have not been implemented.

Examples:
```
     OPEN INPUT CARDS.
     OPEN OUTPUT REPORT-FILE.
```

---

Element: PERFORM

Format:
```
1.  PERFORM procedure-name [THRU procedure
    -name-2]
2.  PERFORM procedure-name [THRU procedure
    -name-2]
       {identifier} TIMES
       {integer   }
3.  PERFORM procedure-name [THRU procedure
    -name-2]
       UNTIL <condition>
4.  PERFORM procedure-name VARYING
    {identifier}
       FROM {identifier} BY {identifier}
       UNTIL <condition>
```

Description:
All four options are supported. Branching may be either forward or backward, and the procedures called may have perform statements in them as long as the end points do not coincide or overlap.

Examples:
```
PERFORM OPEN-ROUTINE.
PERFORM TOTALS THRU END-REPORT.
PERFORM SUM 1Ø TIMES.
PERFORM SKIP-LINE UNTIL PG-CNT GREATER 6Ø.
PERFORM REPEAT-AGAIN VARYING COUNTER FROM
 1 BY 2
   UNTIL COUNTER EQUAL 1Ø.
```

---

Element: READ

Format:
```
1.  READ file-name INVALID <imperative-
    statement>
2.  READ file-name END <imperative-
    statement>
```

Description:
The invalid condition is only applicable to files in a random mode. All sequential files must have an END statement.

Examples:
```
READ CARDS END GO END-OF-FILE.
READ RANDOM-FILE INVALID MOVE SPACES TO
REC-1.
```

---

Element: REWRITE

Format:
```
REWRITE record-name [INVALID <imperative>]
```

Description:
REWRITE is only valid for files that are open in the I-O mode. The INVALID clause is only valid for random files. This statement results in the current record being written back into the place that it was just read from, the last executed read.

Examples:
```
REWRITE CARDS.
REWRITE RAND-1 INVALID PERFORM ERROR-CHECK
```

---

Element: STOP

Format:
```
STOP {RUN    }
     {literal}
```

Description:
This statement stops execution of the program. If a literal is specified, then

the literal is displayed on the console and a prompt is displayed giving the operator the option of terminating or continuing program execution.

Examples:
STOP RUN.
STOP 1.
STOP 'INVALID FINISH'.
For the last two examples the following prompt is displayed:
OPERATOR ENTER A <CR> TO CONTINUE
OR ENTER AN "S" TO TERMINATE.

---

Element: SUBTRACT

Format:
SUBTRACT {identifier-1} [identifier-2] . . . FROM identifier-m
{literal-1 } [literal-2 ]
[ROUNDED] [SIZE ERROR <imperative-statement>]

Description:
Identifier-m, is decremented by the value of identifier/literal one. The results are stored back in identifier-m. Rounding and size error options are available if desired. Multiple subtracts have not been implemented.

Examples:
SUBTRACT 1Ø FROM SUB(12).
SUBTRACT A FROM C ROUNDED.

---

Element: WRITE

Format:
1. WRITE record-name [{BEFORE} ADVANCING {INTEGER}]
{AFTER } {PAGE }
2. WRITE record-name INVALID <imperative-statement>

Description:
The record specified is written to the file specified in the file section of the source program. The INVALID option only applies to random files.

Examples:
WRITE OUT-FILE.
WRITE RAND-FILE INVALID PERFORM ERROR-RECOV.

---

Element: <condition>

Format:
RELATIONAL CONDITION:
{identifier-1} [NOT] {GREATER} {
identifier-2}
{literal-1 } {LESS } {
literal-2 }
{EQUAL }
CLASS CONDITION:
identifier [NOT] {NUMERIC }
{ALPHABETIC}

Description:
It is not valid to compare two literals. The class condition NUMERIC will allow for a sign if the identifier is signed numeric.

Examples:
A NOT LESS 1Ø.
LINE GREATER 'C'.
NUMB1 NOT NUMERIC

---

Element: subscripting

Format: data-name (subscript)

Description:
Any item defined with an OCCURS may be referenced by a subscript. The subscript may be a literal integer, or it may be a data item that has been specified as an integer. If the subscript is signed, the sign must be positive at the time of its use.

Examples:
A(1Ø)
ITEM(SUB)

---

## VI. Error Messages

### A. Compiler Fatal Messages

BR Bad read - disk error, no corrective action can be taken in the program.
CL Close error - unable to close the output file.
MA Make error - could not create the output file.
MO Memory overflow - the code and constants generated will not fit in the allotted memory space.
OP Open error - can not open the input file, or no such file present.
SO Stack overflow - LALR (1) parsing stack has exceeded its maximum allowable size.
ST Symbol table overflow - symbol table is too large for the allocated space.
WR Write error - disk error, could not write a code record to the disk.

---

### B. Compiler Warnings

DD Carriage Control error - The WRITE BEFORE/AFTER ADVANCING option can only be used with sequential files.
CE Close error - attempted to close a non-existing file.
DD Duplicate Declaration - the identifier name has been previously declared.
EL Extra levels - only 1Ø are allowed.
FT File type - the data element used in a read or write statement is not a file name.
IA Invalid access - the specified options are not an allowable combination.
ID Identifier stack overflow - more than 2Ø items in a GO - DEPENDING statement.
IS Invalid subscript - an item was subscripted but it was not defined by an OCCURS.
IT Invalid type - the field types do not match for this statement.
LE Literal error - a literal value was assigned to an item that is part of a group item previously assigned a value.
LV Literal value error - the PICTURE clause field type does not match the VALUE clause literal type.
L7 Level 77 error - level 77 used incorrectly.
MD Multiple decimals - a numeric literal in a VALUE clause contains more than one sign.
NF No file assigned - there was no SELECT clause for this file.
NI Not implemented - a production was used that is not implemented.
NP No production - no production exists for the current parser configuration; error recovery will automatically occur.
NV Numeric value - a numeric value was assigned to a non-numeric item.

OE Open error - attempt to open a file that was not declared; or attempted to open a file for I-O that was not a RELATIVE file.
OL OCCURS LEVEL - Ø1 and 77 levels can not contain an occurs clause.
PC Picture clause - a pic clause exceeds 3Ø characters.
P1 More than one float symbol declared.
P2 Non-numeric data in repetition clause or missing right parenthesis.
P3 Invalid or incompatible symbol in pic clause.
P4 Invalid symbol(s) embedded within a float symbol only /,0,B,',' allowed.
P5 Invalid combination of symbols in pic clause, type cannot be determined.
P6 Number of possible numeric entries exceeds register length max is 18.
PF Paragraph first - a section header was produced after a paragraph header, which is not in a section.
R1 Redefine nesting - a redefinition was made for an item which is part of a redefined item.
R2 Redefine length - the length of the redefinition item was greater than the item that is redefined. That is only allowed at the Ø1 level. This error message may be printed out one identifier past the redefining identifier record in which it occurred.
R3 Redefines misplaced - a redefines was attempted in the FILE SECTION of the source program.
SE Scanner error - the scanner was unable to read an identifier due to an invalid character.
SG Sign error - either a sign was expected and not found, or a sign was present when not valid.
SL Significance loss - the number assigned as a value is larger than the field defined.
TE Type error - the type of a subscript index is not integer numeric.
UD Undeclared identifier - the identifier was not declared.
UL Unresolved label - label has not been referenced. This warning will be given to all references to external subroutines.
VE Value error - a value statement was assigned to an item in the file section.
WL Wrong level error - program attempted to write a record other than an Ø1 level record to an output file.

---

C. Interpreter Fatal Errors

CL Close error - the system was unable to close an output file.
CO Call stack Overflow - insufficient memory available to transfer variable address' and/or return location for a subroutine call.
ME Make error - the system was unable to make an output file on the disk.
NF No file - an input file with the given name could not be opened.
OE Open Error - attempt to open a file which was already open.
OP open Error - the system was unable to open a file.
PS Procedure Stack - not enough memory to load all subroutines.
SO Subroutine Overflow - subroutine symbol table overflow.
W1 Write non-sequential - attempted to WRITE to a file opened for INPUT or a file opened for I-O when ACCESS was SEQUENTIAL.
W2 Wrong key - attempted to change the key value to a lower value than the number of the last record written.
W3 Write input - attempted to WRITE to a file opened for INPUT.
W4 Write non-empty - attempted to WRITE to a non-empty record.
W5 Read output - attempted to READ a file opened for OUTPUT.
W6 Rewrite error - attempted to REWRITE to a file not opened for I-O.
W7 Rewrite error - attempted to REWRITE a record before reading the file; or multiple REWRITE attempts without doing a READ between each.

---

D. Interpreter Warning Messages

EM End mark - a record that was read did not have a carriage return or a line feed in the expected location.
GD Go to depending - the value of the depending indicator was greater than the number of available branch addresses.
IC Invalid character - an invalid character was loaded into an output field during an edited move. For example, a numeric character into an alphabetic-only field.
NE Numeric Error - non-numeric data in an arithmetic operation.
W8 Write Error - the system was unable to write to an output file on the disk. Disk may be full.
SI Sign Invalid - the sign is not a "+" or a "-."

========== THE END ==========

AN IRISHMAN'S LETTER TO THE D.H.S.S

Dear Sir,

I have just received the AIDS leaflet through my door, and would like to apply.

I have been on the dole since arriving from Ireland ten years ago, and am also receiving supplementary benefit and any other state aid I can get. It now seems I am entitled to get aid for sex. It's a pity that this AIDS has come so late in life, as I have already got fifteen children, but I wonder if I am entitled to back payments.

The leaflet states that the more sex I have, the better my chances of getting AIDS. The only problem with that is the wife, who is not so keen on it after 15 kids, and I wondered if you could send someone to talk her into it. To try and encourage her, I bought some sex aids, but these haven't worked. Will I be able to claim the £17.86p I paid for them?

Anyway, I'm sure that when she understands the government will be paying us for sex, she won't let a chance like that slip by.

You state that I can pass my AIDS on to others, but with a wife and 15 kids to feed there wont be much left, but if there is anything, I will give it to my mother-in-law, who only has her pension.

As I can also get AIDS through a blood transfusion, I have written to the hospital to get one. However, I am not sure if this will affect the AIDS I get from you, so could you write and let me know.

I'm a firm believer in the state, as I am sure my past record of claiming all the state aid I can get will show, and I hope this will stand in my favour.

Could you let me know how much I will get each time I have sex, and who keeps the record for payment. Also, will it be paid weekly or monthly?

Yours faithfully.

Shamus O'Toole.

P.S. Your advert is great. I certainly wont die of ignorance, I know my rights.

==========

THE SPARROW

A little sparrow was flying in a bitterly cold wind when suddenly his wings started to ice up solid. Helplessly, he drifted to earth and landed in a barren field.

He started to pray for a quick painless death, when suddenly he was enveloped in a warm covering. A passing cow had decided to drop it's pat exactly on top of the poor little sparrow.

The poor sparow, now enveloped in this warm mixture, soon thawed out and was so happy he decided to sing.

A passing tabby heard the singing and promtly dug out the sparrow and ate it.

There are three morals to this story :-

1. If you are shit on, it's not necessarily your enemy.

2. If you are dug out of the shit, it's not necessarily your friend.

3. If you are in the shit and happy, keep your mouth shut.

==========

```
                BIORYTHM.PAS
              By Mel Saunders
              A Pascal Program
{$T}

PROGRAM BIORHYTHM (INPUT,OUTPUT);
{A Program to calculate your biorhythmic
performance!!
   first version, 14/5/1987.Mel Saunders.}

CONST
        PI=3.145926;
VAR
        DAY1,DAY2,MONTH1,MONTH2,MONTH;
        YEAR1,YEAR2,DUM:INTEGER;
        A1,B1,C1,N,G,D,M,X,Y,Z:INTEGER;
        S,A,B,C,AD,DD:REAL;
        RESPONSE:CHAR;

PROCEDURE MONTHS (M,D:INTEGER);

BEGIN
        X:=TRUNC(D);

        CASE MONTH OF

        2:X:=31+X;
        3:X:=59+X;
        4:X:=90+X;
        5:X:=120+X;
        6:X:=151+X;
        7:X:=181+X;
        8:X:=213+X;
        9:X:=242+X;
        10:X:=273+X;
        11:X:=303+X;
        12:X:=334+X
END
END;

BEGIN
        REPEAT
        DUM:=CPM(2,26);
        WRITELN('BIORHYTHMS':36);
        WRITELN('==========':36);
        WRITELN;WRITELN('Mel Saunders 14th
 April 87.':45);
        WRITELN(        '-----------------
----------':45);
        WRITELN;
        WRITELN('   Biorythms are a way to
detect what kind of STATE you are');
        WRITELN('   in intellectualy,
emotionally and physically. It works on');
        WRITELN('   time cycles of 23,28
and 33 days, that start when you are');
        WRITELN('   first born, and enter
the world!');
        WRITELN;
        WRITE('     Enter your date of
birth like this: DD MM YYYY ');
        READ(DAY1,MONTH1,YEAR1);
        WRITELN;
        WRITE('     Enter todays date in
the same manner ');
        READ(DAY2,MONTH2,YEAR2);

        S:=0;D:=DAY1;M:=MONTH1;Y:=YEAR1;
        N:=TRUNC(365.25*(YEAR2-YEAR1)+30*
(MONTH2-MONTH1)+(DAY2-DAY1));
        MONTHS(M,D);
        AD:=0-X;
        FOR Z:=YEAR1 TO YEAR2 DO
        IF (Z/4-TRUNC(Z/4)=0) AND (Z/100-
TRUNC(Z/100)<>0) AND (YEAR1<>YEAR2) THEN
S:=S+1;
        AD:=AD+S+(365*(YEAR2-YEAR1));
        D:=DAY2;M:=MONTH2;Y:=YEAR2;
        MONTHS(M,D);
        AD:=AD+X+M;
        G:=TRUNC(AD);
        A:=SIN(2*PI*G/23);B:=SIN(2*PI*G/
28);
        C:=SIN(2*PI*G/33);DD:=((A+B+C)/3);
        WRITELN;
        WRITELN('  YOUR INTELLECTUAL LEVEL
 IS....':43,A*99+0.5:6:2);
        WRITELN('  YOUR EMOTIONAL LEVEL
IS.......':43,B*99+0.5:6:2);
        WRITELN('  YOUR PHYSICAL LEVEL
IS........':43,C*99+0.5:6:2);
        WRITELN;
        WRITELN('  YOUR AVERAGE LEVEL
IS.........':43,DD*99+0.5:6:2);
        WRITELN;
        WRITELN('YOU HAVE LIVED FOR APPROX
: ':40,N:5,' DAYS');
        WRITELN;WRITE('PRESS RETURN TO
CONTINUE, CONTROL-C TO END ':52);
        READ(RESPONSE);
        UNTIL RESPONSE=CHR(3)
END.
================ THE END =================
```

AUNTIE DAVIDS PAGE

==========

ZORK TIPS
By Pete Arnold

Zork 1 tips.

1) Have you tried moving the rug ?
2) Don't forget your lamp
3) At the large tree in the clearing try going up!!

Drop me a line if you stuck inside Zork 1, and I will try to solve your problem in the next issue.

Also I can supply a Zork 1 map. Please send a SAE, and it is yours. (A5 minimum size please).

To :- Zork 1 solutions,
Pete Arnold,
18 Owenite Street,
Abbey Wood,
London,
SE2 ØNF.

Thats all for now, more next issue.

Pete....

==========

ZSMALL.DOC
A document file from IUG-2

A SUMMARY OF THE SMALL C LANGUAGE

Introduction
Small C is a subset of the standard C language. It was written by Ron Cain and originally published in Dr.Dobbs Journal of Computer Calisthenics and Orthodontia, No.45. The version described has been modified to generate Z8Ø mnemonics instead of the original 8Ø8Ø ones. A number of bug fixes, most of them sent in by readers of DDJ, have also been incorporated.

The compiler inputs a program written in Small C from a file and produces a version of the program in Z8Ø assembler language mnemonics which can be assembled and run on a Z8Ø system. The original run time library has been supplemented by some additional routines which use CP/M I/O facilities. The library is divided into modules so that routines that are not needed for particular applications can be omitted.

SMALL C SYNTAX SUMMARY

Variable and Function names

All variables must be declared. If they are declared outside a function they are global. Variables and function names should consist of not more than 8 alpha-numeric characters. Global variables and function names appear as labels in the assembler language file produced by the compiler, so they must satisfy any special requirements of the assembler that is to be used.

Permitted Data types

char - 8 bit signed integer with values between -128 and +127 eg. char cin,cout; declares two character variables called cin and cout

int - 16 bit signed integer with values between -32768 and +32767 eg. int num,val; declares two integer variables called num and val

Pointers to char or int data. Pointers contain the addresses of data elements.
eg.
int *size; declares a pointer to an integer
char *pch; declares a pointer to a character

Single dimension arrays of character or integer variables.
eg.
char line[8Ø]; declares a character array of elements line[Ø] to line[79].

Operators
The operators listed below can be applied to expressions rather than just to single variables where appropriate.

1. Unary (These operators group right to left)

- Minus. Forms 2's complement.
* Pointer. Refers to data whose address is given by an expression.
& Gives the address of the expression if any.

++ Increment the expression. ++expression will increment the expression before using it. Expression++ will use the expression then increment it.
-- Decrement. This works like ++.

2. Binary (These operators group left to right except for assignment)

* / Multiply, Divide
% Modulo. Gives remainder after division.
+ - Add, Subtract
>> << Right and left shift. eg. x>>2 will give x shifted right by 2 bits. (***NB*** These are arithmetic shifts: for right shift the highest order bit is propagated into the vacant bit positions)
| Bitwise logical Inclusive OR
& Bitwise logical AND
^ Bitwise logical Exclusive OR
= Assignment of value on the right to the left. a=b=c=5 is a legal expression with a final value of 5.

3. Relational (These also group left to right)

Pointers are unsigned.
Expressions are considered 'true' if their value is non-zero.

== Equal (Unlike the "=" above, this does not cause assignment when used)
!= Not equal
< Less than
> Greater than
<= Less than or equal
>= Greater than or equal

The && and || logical operators of standard C are missing. The bitwise & and | can be used with care between proper logical expressions because these are always given a value of 1 or Ø depending on whether they are true or false.

Types of Statement

N.B. All reserved words such as if, else, while, return etc. MUST be in lower case.

Expression;

if(expression) statement; Statement executed if expression is non-zero

if(expression) statement; Statement executed if expression is non-zero.
else statement; Statement executed if expression is zero.

while(expression)statement; Statement executed while expression is non-zero. (It is possible for the statement not to be executed at all)

break; Control transferred from the innermost while loop

continue; Return control to the top of the while loop

return; return from function

return expression; return from function with value given by expression

; null statement

{statement;statement;statement;..statement;} compound statement which may be used anywhere instead of a simple statement.

Constant values

These can be expressed as:
Decimal numbers within the permitted range.
One or two characters in single inverted commas. eg. 'A' is equivalent to 65 (decimal).
A string in double inverted commas. eg. "Fred Jones" this has a 'value' consisting of a pointer to the string, terminated by a null byte, somewhere in memory.

Pseudo-preprocessor Statements

£define name string Replace name by string throughout the program
£include filename Insert named file at this point in the program
eg.£include CRUN.LIB Included files may not contain £include statements.
£asm ... £endasm This allows assembly language to be included in a program. Anything between £asm and £endasm is passed straight to the O/P file.
Comments are written: /* Comment */

The Run Time Libraries

Small C programs should start with a statement: "£include CRUN.LIB". The run time routines in CRUN.LIB include an inital section which will set up the stack and then call the function "main()" which must exist in all C programs. The group of routines in CRUN.LIB are essential in all Small C programs, except for the "getp" and "putp" functions. The other three libraries can be £included if required. CONIO.LIB contains functions that will normally be needed for CP/M character and string I/O from the console, FILE.LIB (which assumes CONIO.LIB is also present) can be £included if files are going to be used. NUMIO.LIB contains some routines in C for inputting and outputting decimal and hexadecimal numbers.

To keep the management of file channels simple, only one input and one output file may be open at a time. CON: and LST: are acceptable O/P names. After compilation the stack pointer setting and the ORG pseudo-op can be set to suit a particular application if required.

The library routines are quite heavily commented. To speed up the processing of £included files, it may be worth preparing uncommented versions.

If software for a stand alone system is being written, the run time routines can be reduced to the group of essential routines in CRUN.LIB plus the "getp" and "putp" functions. As far as is known, the compiler will run on an 8Ø8Ø so it should be useable in conjunction with an assembler such as Z8ØASM by 8Ø8Ø/8Ø85 systems. In the run time library the "getp" and "putp" functions use Z8Ø instructions so these would have to be changed for 8Ø8Ø/8Ø85 applications.

Functions in CRUN.LIB

getp(n) will get an 8 bit value from port n. (Z8Ø only)
putp(ch,n) will output an 8 bit value to port n. (Z8Ø only)

(The rather more obvious function names "in" and "out" don't work with Z80ASM as they clash with Z80 mnemonics.)

The file starts with default settings of: ORG 100H and LD SP,9000H. The rest of the file consists of essential utility routines from the original library published by Ron Cain.

Functions in CONIO.LIB

| | |
|---|---|
| cpm(DE_value,C_value) | makes a CP/M system call. Returns with value of A. |
| * getchar() | Get a character from the keyboard. (Echoed to screen) |
| * putchar(c) | Put a character to the VDU |
| * gets(buff) | Get a null terminated string into a buffer pointed to by buff |
| * puts(buff) | Output a null terminated string pointed to by buff to VDU |
| \n,\b and \t | Give newline,backspace and tab. |
| crlf() | Output carriage return and line feed to VDU |
| lston() | Turn on LST: device for CON: output |
| lstoff() | Turn off LST: |

Functions in FILE.LIB

| | |
|---|---|
| * putc(char,iochannel) | Output character to opened output file. Returns character value or -1 if error. |
| * getc(iochannel) | Get character from opened input file. Returns a value or -1 if error. |
| * fopen(filename,mode) | eg.io= fopen("FRED.DAT", "r"); use "r" for read files and "w" for write files. The file name is passed as a pointer. LST: and CON: are acceptable output file names. The function returns the value of the iochannel assigned, or zero if the open fails. Only 1 "r" and 1 "w" file may be open at a time. The channels assigned in this version are always 1 for "w" and 2 for "r". |
| * fclose(iochannel) | Close a file. (fclose(0) causes the closing of the only write file without a terminating ^Z.) |
| eof() | This returns a value of 0 until an attempt has been made to read a sector beyond the end of a file. ASCII files will normally have returned ^Z before eof() becomes 'true'. |
| * exit() | Exit C program, close write files, and do a warm start. |

Functions in NUMIO.LIB

| | |
|---|---|
| getdec() | Get a decimal number from the keyboard with a '?' prompt |
| putdec(n) | Outputs a signed decimal number to the VDU |
| gethex() | Gets a hexadecimal number from the keyboard with '$' prompt |
| putbyte(ch) | Output a character as a 2 digit hex number |
| puthex(n) | Output an integer as a 4 digit hex number |

[* indicates a function which is similar (but not always identical) to one in standard C.]

Using the Small C Compiler.

Use a normal editor to write your program, no particular file type is assumed though .C is usual. Run the compiler. You will then be asked several questions. It is usually a good idea to ask for your C program to be included in the assembler file as comments. Global variables will normally need to be defined, unless a very large program is being compiled into several separate assembler files. Choose a small number for the start of the label numbering. Then give the names of the output and input files. Names must be given in full and may include a drive name. LST: and CON: are acceptable if you just want a compiler output listing. When compilation is complete you will be asked for another input file name which can be added to the one compiled previously. Usually you just enter a <RETURN> to exit from the compiler. It is possible to enter <RETURN> as the answer to the first three questions on entering the compiler, in which case the program will default to including the C program as comments, definition of globals and labels starting at 0. You may have to do some editing to your assembler language file. You will find that global variables have been given storage at the end of the listing. If the program is intended for a stand-alone application with its final code in EPROM you will have to put in another ORG pseudo-op before the variable addresses to locate them in RAM. ROMable code may only use CRUN.LIB, the other LIB files all use buffers,flags etc which have to be in RAM or rely on routines that do. For normal CP/M programs editing shouldn't be needed. When the file has been compiled, assemble the output file in the usual way and run it.

Linking Small C output to machine code

If a function is called: eg. jim(arg1,arg2,arg3), the compiler pushes the values of arg1,arg2 and arg3 on the stack in the order that they are listed and then generates a call to a label "jim". If an argument is a string such as "Fred Jones", the argument value will be a pointer to the string which will be somewhere in memory terminated by a null byte. Values should be returned in HL. 8 bit values should be sign extended to 16 bits and the simplest way to do this is to put the value into A and jump to a routine in CRUN.LIB called "ccsxt" which will do this for you. The stack must be restored to its previous state before returning. There are examples of this in the library files. Machine code functions can be put between £asm and £endasm directives among the Small C functions or they can be £included from a separate file (also between £asm and £endasm).

Some other details of the compiler

The original Small C program published by Ron Cain was itself written in C. The version described here was typed into an Onyx in the Electrical Engineering Department at Brighton Polytechnic and compiled on the Onyx's C compiler to produce a Small C compiler running under UNIX. A slightly modified version of the Small C source was then fed into the Onyx's Small C compiler to produce a Z80 assembler file which was assembled. The final Z80/8080 code was then moved to a microcomputer and hence to a floppy disc. This version has been further modified under CP/M. As far as is known the slight eccentricities of Z80ASM have been accomodated. The output is also compatible with other normal Z80 assemblers.

In the compiler support routines, 1k buffers are used for the read and £include input streams and the normal buffer at $80 for output. The £include routine in the compiler also passes an argument "i" to the fopen routine so as to distinguish the call from the read fopen call which passes the standard "r".

Parameters are passed to these routines on the stack. My routines are similar to the ones in CONIO.LIB and FILE.LIB except for the larger i/p buffers and the possibility of "i" mode.

The compiler uses memory up to nearly $8000 so that it will need a system with at least 40k of memory to run in. The stack setting in CRUN.LIB has been set rather arbitrarily at $9000.

The compiler source is divided into two sections ZSC-1.C and ZSC-2.C. They can be combined into a single file if desired, but I find them easier to edit in their present form. If it is desired to recompile the compiler just run ZSMALL.COM and enter first ZSC-1.C as the input file and then ZSC-2.C when the name of the next input file is asked for. The support library ZSC-COMP.LIB is £included in ZSC-1.C and must be on the current drive during compilation. The assembler file of the compiler without C source included as comments is about 128 kbytes in length.

References.

Dr. Dobbs Journal of Computer Calisthenics and Orthodontia, Box E, Menlo Park, CA 94025

No. 45 (May 1980) "A Small C Compiler for the 8080's" by Ron Cain A listing in C of the 8080 compiler + notes on implementation. This issue also contains other articles about C.

No. 48 (Sept. 1980) "A Runtime Library for the Small C Compiler" by Ron Cain Run time library + useful notes. (See also Nos. 52, 57 and 62 for bug notes by P.L.Woods, K.Bailey, M.Gore & B.Roehl, J.E.Hendrix)

Nos. 81 and 82 (July, Aug. 1983) "RED - A better Screen Editor" by E.K.Ream A screen editor written in Small C with a number of library routines.

There is a later version of Small C by J.E.Hendrix in DDJ Nos.74 and 75 but this is copyright and so presumeably not in the public domain.

- - - -

"The C Programming Language" by B.W.Kernighan and D.M.Ritchie (Prentice-Hall) The standard text on C.

The C/80 compiler from Software Toolworks is an 8080 code compiler which seems to be based on Small C but is considerably enhanced.

John Hill (Nov.1983/Mar.1984)

========== THE END ==========

CENTHEAT
by P.J.ARNOLD
For Microsoft Basic

This program was written to allow the calculation of the heat needed in Watts for each room of a typical house.

To use please use only metric units of measurement i.e. Metres or Millimetres, When inputting measurements of more than 1 window in a wall just add them together to make one big window, also input outside doors as single glazed windows (adding them to window totals) and ignore inside doors altogether.

The program is menu driven at each stage and it is quite difficult to go wrong!!, if at any stage you want to bypass a question just input a "Ø" (zero) as the measurement.

It is probably best to write down all your measurements on to a piece of paper and input the lot in one go, The measurements that the program needs are:- height and length of each wall, height and width of all windows in a wall, length and width of floor (the ceiling is taken as being the same as the floor).

The other parameters that the program needs are whether the floor is ground or first, suspended wood or solid, and whether a ceiling is intermediate or roof level.

After this lot has been entered for EACH ROOM the program first calculates the heat loss for each wall and outputs the results to the printer, then the program calculates the total heat loss for the room and this is then output to the printer as well, This figure being the size of radiator or radiators that are needed to heat the room, this carries on for each room in turn until all rooms have been covered that you want to have heated.

The program then calculates the boiler size needed for heating and the boiler size needed to cope with heating and hot water, these figures being output to the printer.

I have taken some liberties with the calculations but when I checked the program against a professional calculator my program was always within a few hundred watts of the professional calculator.

There is no error trapping in the program but you usually know when you have made a mistake because you end up with a massive rad size or a boiler that would not go amiss in the QE2.

If you have any improvements please send them to Interaktion, I would like to see them.

Bye for now PETE.....

```
1Ø REM CENTHEAT BY P.J.ARNOLD
2Ø ?CHR$(26)
3Ø DIM R(11,1Ø),U(1Ø,11),TT(1Ø),TL(4)
4Ø ?:?:?:?:?"Central Heating Calculator"
5Ø ?:?"Please use only Metric units of mea
   surement"
6Ø ?:?:INPUT"Input Name:- ";AZ$
63 ?:INPUT"Input house No & street:- ";AT$
64 ?CHR$(
8Ø INPUT"How many rooms ? ";S
1ØØ FOR Y=1 TO S
11Ø ?CHR$(26)
112 ?:?:?
12Ø ?"**********************************
   ******"
13Ø ?" Input room to be calculated-------
   -Enter"
14Ø ?"----------------------------------
   ------"
15Ø ?"Dining room ----------------------
   >  A "
16Ø ?"Living room ----------------------
   >  B "
17Ø ?"Kitchen --------------------------
   >  C "
18Ø ?"Hall -----------------------------
   >  D "
19Ø ?"Bedroom 1 ------------------------
   >  E "
2ØØ ?"Bedroom 2 ------------------------
   >  F "
21Ø ?"Bedroom 3 ------------------------
   >  G "
22Ø ?"Bedroom 4 ------------------------
   >  H "
23Ø ?"Bathroom -------------------------
   >  J "
24Ø ?"Toilet ---------------------------
   >  K "
25Ø ?:?:?:INPUT A4$
26Ø IF A4$="A" THEN A3$="Dining room":
   RT=21:RR=1
27Ø IF A4$="B" THEN A3$="Living room":
   RT=21:RR=2
28Ø IF A4$="C" THEN A3$="Kitchen":RT=18:
   RR=3
29Ø IF A4$="D" THEN A3$="Hall":RT=16:RR4
3ØØ IF A4$="E" THEN A3$="Bedroom 1":RT=18:
   RR=5
31Ø IF A4$="F" THEN A3$="Bedroom 2":RT=18:
   RR=6
32Ø IF A4$="G" THEN A3$="Bedroom 3":RT=18:
   RR=7
33Ø IF A4$="H" THEN A3$="Bedroom 4":RT=18:
   RR=8
34Ø IF A4$="J" THEN A3$="Bathroom":RT=22:
   RR=9
35Ø IF A4$="K" THEN A3$="Toilet":RT=21:
   RR=1Ø
36Ø GOSUB 1Ø2Ø
37Ø ?CHR$(26)
38Ø DIM AS(4)
39Ø AS(1)=R(RR,2)-R(RR,6)
4ØØ AS(2)=R(RR,3)-R(RR,7)
41Ø AS(3)=R(RR,4)-R(RR,8)
42Ø AS(4)=R(RR,5)-R(RR,9)
43Ø IF R(11,1)=1 THEN TD=2
44Ø IF R(11,1)=2 THEN TD=(RT+3)
45Ø TA=AS(1)*U(RR,2)*TD
46Ø TW=R(RR,6)*U(RR,6)*TD
47Ø TL(1)=INT(TA+TW)
471 LPRINT:LPRINT
472 LPRINT CHR$(9);
473 LPRINT CHR$(14);"Central Heating Calcu
   lations"
474 LPRINT:LPRINT CHR$(9);CHR$(9);
475 LPRINT"for ";AZ$;" of ";AT$
48Ø LPRINT:LPRINT:LPRINT A3$
49Ø LPRINT:LPRINT"The total fabric loss
   for wall 1 =----->";TL(1);" Watts"
5ØØ IF R(11,2)=1 THEN TD=2
52Ø TA=AS(2)*U(RR,3)*TD
53Ø TW=R(RR,7)*U(RR,7)*TD
54Ø TL(2)=INT(TA+TW)
55Ø LPRINT:LPRINT"The total fabric loss
   for wall 2 =----->";TL(3);" Watts"
56Ø IF R(11,3)=1 THEN TD=2
57Ø IF R(11,3)=2 THEN TD=(RT+3)
58Ø TA=AS(3)*U(RR,4)*TD
59Ø TW=R(RR,8)*U(RR,8)*TD
6ØØ TL(3)=INT(TA+TW)
61Ø LPRINT:LPRINT"The total fabric loss
   for wall 3 =----->";TL(3);" Watts"
62Ø IF R(11,4)=1 THEN TD=2
63Ø IF R(11,4)=2 THEN TD=(RT+3)
64Ø TA=AS(4)*U(RR,5)*TD
65Ø TW=R(RR,9)*U(RR,9)*TD
66Ø TL(4)=INT(TA+TW)
```

```
670 LPRINT:LPRINT"The total fabric loss
    for wall 4 =----->";TL(4);" Watts"
680 IF R(11,5)=2 THEN GOTO 740
690 FL=INT(R(RR,10)*U(RR,10)*RT+3)
700 LPRINT:LPRINT"The total fabric loss
    for floor =------>";FL;" Watts"
710 CL=INT(R(RR,10)*U(RR,11)*2)
720 LPRINT:LPRINT"The total fabric loss
    for ceiling =---->";CL;" Watts"
730 GOTO 780
740 FL=INT(R(RR,10)*U(RR,10)*2)
750 LPRINT:LPRINT"The total fabric loss
    for floor =------>";FL;" Watts"
760 CL=INT(R(RR,10)*U(RR,11)*(RT+3))
770 LPRINT:LPRINT"The total fabric loss
    for ceiling =---->";CL;" Watts"
780 VX=R(1,1)*R(RR,10)
790 IF RR=1 THEN AA=2
800 IF RR=2 THEN AA=2
810 IF RR=3 THEN AA=3
820 IF RR=4 THEN AA=1.5
830 IF RR=5 THEN AA=1.5
840 IF RR=6 THEN AA=1.5
850 IF RR=7 THEN AA=1.5
860 IF RR=8 THEN AA=1.5
870 IF RR=9 THEN AA=2
880 IF RR=10 THEN AA=2
890 VC=VX*AA*(RT+3)*.37
900 TT(RR)=INT(TL(1)+TL(2)+TL(3)+TL(4)+CL+
    FL+VC)
910 LPRINT:LPRINT"The total loss for the
    room is ";TT(RR);" Watts"
912 LPRINT:LPRINT:LPRINT
920 NEXT Y
930 BT=TT(1)+TT(2)+TT(3)+TT(4)+TT(5)+TT(6)
    +TT(7)+TT(8)+TT(9)+TT(10)
940 BZ=BT*10/100
950 BV=INT(BZ+BT)
960 LPRINT:LPRINT"Total Boiler load for
    central heating = ";BV;" Watts"
970 BS=INT(BV+4000)
980 LPRINT:LPRINT"Total Boiler load with
    domestic hot water = ";BS;" Watts"
985 ?CHR$(26)
990 ?:?:?:?:INPUT"Finished with calculator
     Y/N ";A1$
1000 IF A1$="N" THEN GOTO 90
1010 IF A1$="Y" THEN END
1020 REM MAIN I/P ROUTINE
1030 ?CHR$(26)
1040 ?:?
1050 INPUT"Input height of walls:-";R(1,1)
1070 FOR I=1 TO 4
1090 ?:?"Enter length of wall:-";I
1110 ?:?:INPUT A2:R(11,I+4)=A2
1120 R(RR,I+1)=A2*R(1,1)
1130 ?CHR$(26)
1140 ?:?:INPUT"Is wall external or
    internal ? E/I ";A1$
1150 IF A1$="I" THEN U(RR,I+1)=2.3:
    R(11,I)=1:GOTO 1330
1170 IF A1$="E" THEN R(11,I)=2
1180 INPUT"Are walls solid or cavity ?
    S/C ";E$
1190 IF E$="S" THEN U(RR,I+1)=2.4:
    U(1,1)=2.4:GOTO 1240
1210 ?:INPUT"Is cavity insulated Y/N ";E$
1220 IF E$="Y" THEN U(RR,I+1)=.5
1230 IF E$="N" THEN U(RR,I+1)=1.9
1240 ?CHR$(26)
1241 ?:?:INPUT"Does this wall contain
    windows Y/N ";A1$
1242 IF A1$="N" GOTO 1330
1243 ?CHR$(26)
1250 ?:?:?:INPUT"Input total length of
    windows:-";W1
1260 ?
1270 INPUT"Input height total height of
    windows:-";W2
1280 ?
1290 INPUT"Are windows double or single
    glazed ? S/D";A1$
1310 IF A1$="S" THEN U(RR,I+5)=5.6
1320 IF A1$="D" THEN U(RR,I+5)=3.4
1330 ?CHR$(26)
1340 R(RR,I+5)=W1*W2
1350 W1=0:W2=0
1360 NEXT I
1400 ?:?:?:INPUT"Input length of floor:-"
    ;LF
1410 ?
1420 INPUT"Input width of floor:-";WF
1430 R(RR,10)=LF*WF
1440 ?CHR$(26)
1450 ?:?:?:INPUT"Is the floor on the first
     or ground floors G/F ";A1$
1470 IF A1$="F" THEN U(RR,10)=1.2:
    R(11,5)=2:GOTO 1550
1480 IF A1$="G" THEN R(11,5)=1
1490 ?CHR$(26)
1500 ?:?:?:INPUT"Input is floor suspended
    wood or solid  S/W ";A1$
1520 IF A1$="S" THEN U(RR,10)=1.1
1530 IF A1$="W" THEN U(RR,10)=1.4
1550 ?
1560 INPUT"Is ceiling intermediate or roof
    level I/R ";A1$
1580 IF A1$="I" THEN U(RR,11)=1.6:
    GOTO 1630
1590 IF A1$="W" THEN U(RR,11)=2.2
1600 ?
1610 INPUT"Has roof been insulated Y/N "
    ;A1$
1620 IF A1$="Y" THEN U(RR,11)=.5
1630 ?CHR$(26)
1640 RETURN

================ THE END =================
```

EVANS ABOVE

Z8ØDOCUK.DOC
A document file from IUG-2

Z8ØASM - ASSEMBLY LANGUAGE PROCESSOR.

Z8Ø-CPU ZILOG/MOSTEK MNEMONICS

LATEST REVISION: 1-SEPTEMBER-1982

Z 8 Ø A S M

Z8ØASM is an assembler for the Z8Ø-CPU Microprocessor, using the ZILOG/MOSTEK Mnemonics. It is designed to run under the CP/M 1.4 or 2.2 Operating systems from Digital Research. CP/M will run on either an 8Ø8Ø or a Z8Ø system. Z8ØASM will run in the minimum CP/M system (16k) but will be able to use more memory for symbol table storage in larger systems (up to 64k).

Z8ØASM reads a source (.ZSM) file created using the CP/M 'ED' or similar Text Editor. The source may be in either Upper or Lower case letters. The program produces an optional Object code (.HEX) file in INTEL HEX format, and a listing sent either to the Console (CON:) device or the (LST: device), or to a print (.PRN) file. The 'HEX' file can be loaded for execution using 'DDT' or 'LOAD' utilities of CP/M or it can be punched on to tape or sent down a modem using 'PIP' or 'BSTAM'.

Input statements are free-format (i.e NOT column oriented). Between fields any number of blank or tab characters may be present, but within a field there may be no blank or tab characters. However, statement labels must be in the first position of the line.

The first character of a label MUST be a letter, with following characters being any alpha-numeric string. Statement labels should be followed by a colon. Comments are preceeded by a semi-colon and may appear by themselves or following all of the fields on a source line.

All the ZILOG/MOSTEK Mnemonics are supported with the following notes:

1. The syntax of ROTATE and SHIFT instructions is compatible with both 8Ø8Ø and Z8Ø patterns. eg. 'RLCA', 'RRA', 'RR A', 'SL A' etc.

2. The 8-BIT IMMEDIATE arithmetic instructions are accepted in the two alternative forms :-

```
ADD <EXPRESSION>
ADD A,<EXPRESSION>  etc.
```

3. Index register displacement values MUST be stated even if the value is Zero. The MOSTEK manual implies that an unspecified zero displacement is illegal, so this should be declared as LD A,(IX+Ø, though some Z8Ø Assemblers do allow LD A,(IX).

The list below describes the PSEUDO-OPERATORS available in Z8ØASM and their argument formats:

```
<LABEL> EQU  <EXPRESSION> ASSIGN VALUE TO
                          LABEL
        ORG  <EXPRESSION> ASSIGN VALUE TO
                          PROGRAM COUNTER
        DEFS <EXPRESSION> RESERVE STORAGE
        DEFW <EXPRESSION> DEFINE WORD (2
                          BYTES)
        DEFB <EXPRESSION> DEFINE BYTE
                          (FORMAT 1)
        DEFB '-STRING-'   DEFINE BYTE
                          (FORMAT 2)
        EJECT             LISTING EJECT
        END  <EXPRESSION> DEFINE EXECUTION
                          ADDRESS
```

NOTE: for DEFW only one Operand is allowed. While for DEFB mixed <EXPRESSION> and 'STRINGS' separated by commas are permitted.

EXPRESSIONS:

Expressions may be any number of the items listed, separated by + - * and / for addition, subtraction, multiplication and division respectively.

The Division supported is an unsigned 16-bit by 8-bit integer divide with the Quotient only being returned, theremainder is discarded.

Brackets within expressions are not permitted, but if used will be flagged as errors.

BOOLEAN Operators are NOT permitted, though if used these probably will not be flagged as an error, and the statement will not be assembled correctly.

EXPRESSION ELEMENTS:

```
<LABEL>    1 TO 11 CHARACTER STATEMENT
           LABEL
<NUMBER>H  HEXADECIMAL NUMBER
<NUMBER>O  OCTAL (BASE 8) NUMBER
<NUMBER>Q  OCTAL (BASE 8) NUMBER
<NUMBER>B  BINARY (BIT) NUMBER
<NUMBER>   DECIMAL NUMBER
   $       PROGRAM COUNTER REFERENCE
  'X'      X = ANY PRINTABLE CHARACTER
  'XX'     S ABOVE, BUT FOR 16-BIT
           REGISTER
```

EXAMPLES OF EXPRESSIONS ARE:

```
LABEL1+5
ØFFH
42Q
2ØØ
1Ø1ØØØØ1B+6
$-6
'9'+1
LD HL,25*1Ø
LD A,44/11
LD HL,'AB'
```

The source file must have an extension of 'ZSM'. The object file will have the extension 'HEX' and the print file that of 'PRN' if they are created.

Z8ØASM is called using a command with the following format:

```
Z8ØASM <FILENAME>.<ABC>

or Z8ØASM n:<FILENAME>

n:          OPTIONAL DRIVE NAME, FOR
            DEFAULT OF ALL FILES
<FILENAME>  1 TO 8 CHARACTER CP/M FILE
            NAME (MUST BE .ZSM)
<ABC>       OPTION CONTROLS-
            A=A TO D FOR DRIVE NAME OF
               SOURCE FILE
            B=A TO D FOR DRIVE NAME OF HEX
               FILE
            =Z TO SKIP GENERATION OF A HEX
               FILE
            C=A TO D FOR DRIVE NAME FOR
               PRN FILE
             =P FOR LISTING TO LST: DEVICE
```

=X FOR LISTING TO CON: DEVICE
=Y FOR LISTING TO CON: DEVICE (ERRORS ECHO ON LST:)
=Z TO SKIP LISTING (ERRORS APPEAR ON CON: DEVICE)

IF <FILENAME> ONLY OR n:<FILENAME> FORMAT IS USED, THE A, B AND C OPTIONS DEFAULT TO THE DEFAULT DRIVE
NOTE: Even if the listing device is not requested, any errors that occur will be listed on the CON: device.

ERROR MESSAGES:

These are displayed in the FIRST column of the output line in which they occur, just before the instruction address field.

They consist of:

E = DEFB Expression badly defined
L = Label too long (> 11 chars)
M = Multi-defined label
O = Opcode error (usually mis-spelled), or illegal statement
U = Undefined label
V = Relative displacement out of range

========== THE END ==========

ADVT.DOC
A document file from IUG-2

This is an expanded version of Adventure. This one has a cave that is twice as large (it's a 550 point version). All of the features of the original adventure are in this version, plus a whole bunch of new rooms, treasures, and ways for the bumbling explorer to get oneself killed.

Mike Goetz (NYACC) who adapted the original version of Adventure for CP/M (SIG/M volumes 1, 2, and 3) has released this expanded version to the SIG/M group for non-commercial distribution.

========== THE END ==========

LETTERS

J.D Ritchie,
S.T.E.F.
R.A.F. Finningley,
Nr Doncaster,
S.Yorks,
DN9 3LQ.

Dear Bob,

The mere fact that you are reading this letter is an indication of a significant advance in the development of my Interak system. To wit the addition of a printer! At long last, and I am talking of a period in excess of five years since I began computing, I am in a position to write letters, take listings and generally free myself from the drudgery associated with recording my computing activities by hand.

At the moment the printer is a marvellous new toy and while I would not wish to deny myself the pleasure that goes with playing with toys I hope that I will soon accquire the expertise that will change the toy into a powerful and important tool.

Further to that, how on earth do you enter the printer control codes? I've spent some considerable time trying to change the print out from 12 to 10 cpi and I cannot seem to find the magic combination. If I cannot cope with this there doesn't seem much hope of my being able to manage the more advanced stuff! I've tried the ESC sequences the CTRL sequences and entering a short basic program but nothing seems to work. If I type CTRL [ P the program goes into Printer mode. If I try CTRL [ M I'm told I don't have MailMerge! AAAARRGGGHHH!!!

Short of writing an assembler program to initialise the printer, and there is no guarantee that this will work either, I am at a loss. HELP!! Oh one piece of information you may find useful, the printer is an FX 800 purchased from those excellent gentlemen at Greenbank.

Well, as I have suceeded in preparing and printing this letter I will now write an open letter for the newsletter.

Floreat Interak!
T.T.F.N.
John.

[Ed - Please disable NLQ at the DIP switch before following my suggestions, as the power on NLQ DIP selection sometimes screws up the other modes of operation. (I will show you how to soft-select NLQ later). Power the printer off then on to reinitialise it into draft mode.

The CPMUGUK provide through there library a program called ANYCODE. This when added to Wordstar will alow hex sequences to be embedded into a document file.

I think you have it installed already and we can find out by a simple test.

First some info on ANYCODE.

When it is installed, the installer decides on two character codes that are never to be used as printable characters.

One is declared as an ESC marker and the other is declared as a Hex marker. In your case this will be as follows :-

Tilde mark. (the sine wave symbol. Decimal 126 or shifted ^). When entered this tells Wordstar to pass ESC (27) to the printer.

Back quote. (Decimal 96 or shifted @, or the opposite symbol to '), tells Wordstar to pass the ASCII pair that follows to the printer as one byte. The only restriction is that upper case must be used for the Hex pattern.

Of course I cannot print the tilde symbol as it means escape. I will write this key press as s^ for now but remember I really mean shifted ^.
Also, I cannot type the backwards quote symbol and so I will use s@ which means shifted @ should be pressed. Try this :-

Create a file as :-

AAAA BBBB CCCC DDDD EEEE FFFF GGGG HHHH

Print it and then edit the file to be :-

s^s@4D
AAAA BBBB CCCC DDDD EEEE FFFF GGGG HHHH

Print this and it should be in Elite pitch.
The s^s@4D causes Wordstar to pass the sequence ESC 4D to the printer or ESC M if you like, which forces the printer into Elite.

If this works, any sequence can be transmitted to the printer, with the loss of the ability to print tilde and back quote.

If the previous example worked try :-

s^s'4D'47
AAAA BBBB CCCC DDDD EEEE FFFF GGGG HHHH

To put the printer in Double strike Elite mode, by sending :-

ESC 4D  Elite
47  Double strike

NLQ can be turned on by using the sequence

s^s'78'31
AAAA BBBB CCCC DDDD EEEE FFFF GGGG HHHH

s^s'78  = ESC x
'31  = ASCII 1

To send ESC x 1 to the printer.

If all of this doesn't work and if you send me a disk, I will pass ANYCODE on to you for installation into Wordstar. It is quite simple to do

Finally, if you could let me have your work "on disk" it will help me no-end and I will return your disk post haste. Usually with somthing on it that has come my way.

Good luck and please write and tell us what happens.]

**********

Bruce Joyce,
38 Castle road,
Salisbury,
Wiltshire,
SP1 3RJ.

Dear Bob,

Thank you for including one of my programs in the previous addition of Interaktion. Unfortunatly it highlighted a problem that I had not previously considered.

I have written many programs, usualy games, and many of them include graphics. Stars was no exception. The trouble is that when it is listed on the printer then any inverse characters are left as blanks. Therefore it would have been impossible for any-one to type in accurately from the listing.

I would propose therefore a standard to be adopted for any graphical contributions. Using Crystal basic or Zybasic(cpm) it would be possible to write these statements out using chars$ so we would know what was being written. This of course could be done in a REM statement if it would slow down the program too much. The other way would be to avoid special characters all together. I have enclosed a copy of the missing lines of stars and hope they are of use to someone.

If any one would like a copy of my programs then please write to the above address. There are at least ten that are good (ask David Parkins) and are in Crystal basic with 64 character screen and on tape.

Now on to CP/M.

I am currently writing a program using Zybasic and what a pleasure it is to be able to save and load so easily. I have set up a Profile.Sub on my Zybasic disk that starts up the basic as soon as I boot.

I find Zybasic a very easy but good implementation of the language. I do though have two requests. Is there any chance of having string arrays (multi) added to it and also would it be possible to manipulate files from within a program.

My computer is just a hobby. My desire is to build up a set of buisness programs (databases ect). Of course I realise that it is possible to buy them but the object of the exercise is for me to 'play' with my computer. So is there any chance of adding the extras previously mentioned?

I am just completing a program for Zybasic based on the television program 'Blockbusters' but it would have been easier if string arrays were avaliable. When it is complete it will be freely avaliable to all on CP/M.

Well thank you Bob for all your work and I will enclose the offending lines from 'Stars'.

Yours faithfully
Bruce
(O.K. Beam me up Scotty)
P.S. How about Veltext on CP/M Pete?

MODS TO 'STARS' (NOTE! I HAVE A BETTER VERSION WITH SOUND FOR ANYONE WHO WOULD LIKE A COPY)

```
28Ø IFKBD=32THENPRINT@C,D;CHAR$(&1ØØ):
    LETSCR=SCR+1:PRINT@55,5;2ØØ-SCR
4ØØ PRINT@2Ø,5;"*****************
41Ø PRINT@2Ø,6;"*   STAR TRAP   *
42Ø PRINT@2Ø,7;"*       by      *
43Ø PRINT@2Ø,8;"*  BRUCE JOYCE  *
44Ø PRINT@2Ø,9;"*****************
55Ø PRINT[11 * CHR$(&2Ø)] [1Ø * CHR$(&1ØØ)
    ] [56 *CHR$(&2Ø)] [3* CHR$(&1ØØ)]
    [CHR$(&2A)] [3 * CHR$(&1ØØ)
56Ø PRINT[16 * CHR$(&2Ø)] [CHR$(&1ØØ)]
```

FINAL NOTE! CRYSTAL SAY THAT THERE IS AN UPDATE FOR XTAL DISK CPM FROM THE USER GROUP. HOW MUCH AND HOWE, PLEASE!

[Ed - Thanks for the letter Bruce. Sorry I missed getting it into number 16. First there is no Crystal update from the User Group. However if you give Crystal a ring there is a CP/M version that you can buy. You will probally have to self install it for the VDU2K control sequences but that should present no real problem. As to ZYBASIC. I intend to add further functions as time goes on, but I have been extremly busy of late. Newsletter number 16 was only just completed in time for publication. Whenever I add functions to ZYBASIC, it seems to require two to three months of programming effort, spare time of course. After each such session I need to rest from it for a while as I become stale and edgy. I will be extending the program but you will have to wait until I can muster the grit that is required to attack it, sorry but it is quite a difficult program to cut and paste.
As to a standard for control sequences. Yes I do think we should adopt a method. One good way that was recently used by Paul M.Nicklin, (IUGN-15 page 14), was to give the control key sequences in square brackets such as :-

```
41Ø P."[CTRL_S],[9*CTRL_Y]+N$,Y,[CTRL_P]
```

This struck me as very readable as :- Control and S, 9 times control and Y, N string, Y as a number followed by a control and P. Perhaps this method could be adopted.]

**********

Chris Korycinski,
17 Pitcullen Terrace,
Perth,
PH2 7EQ. 13th May 1987

Dear Tom,

Interaction News Letter.

I have been receiving the newsletter for over a year now. Initially I requested it to find out whether I wanted to actually go ahead and build an Interak computer. As events in the microcomputer world have moved on past CP/M (and the Interak can't even run that adequately - no 8Ø column screen) and into MS-DOS and now OS/2, I really don't see how I could justify either the time or money involved in making Interak.

I think that if the CP/M system had been available in 1983, then it would have been a different story, but as it is I should be grateful if you could stop sending me the newsletter.

Yours sincerely

Chris Korycinski

[Reply by Tom Evans. Sysop and sec. 18/5/87

Hello Chris,

sympathies with your feelings about the Interak and modern O/S systems, but really it is all down to imagination with what you can do with the basic Interak. I have been running Interak complete with 8Ø column terminal, and CP/M now since mid 1984, this has been operating dBase2, Wordprocessors, and accounts very efficiently, and is still doing so, I also run an IBM PC which is "SLOWER" than the Interak when they are running the same stable products ie, dBase2, Wordprocessor, these are the apropriate versions for CP/M and PCDOS. The IBM PC is now acting as Interak's grapics terminal, well now I have the IBM, I must find a use for it! Getting back to 8Ø columns, I used an ADDS Regent terminal that I picked up from a surplus dealer for around 125 quid, but I didn't then need to fork out for the keyboard, and VDU card for Interak (although I had them, lack of forsight originally), you can now pick up terminals for around 5Ø quid, cheap eh! If you want colour as well, this can also be hooked up to the Interak, this again I have done with a Pluto Graphics card, this in conjuction with the Interak knocks the pants of the IBM when it comes to speed and graphics capability.

The OS/2 system proper is not quite ready yet, and according to IBM it will be ready at the end of this year, but according to the authors, it may be a year+ away for a fully de-bugged version, add this to the problem Intel has found with the 836 chip, and fings ain't so Rosey.

I think the OS/2 hardware is at present running on a modded MSDOS O/S.

I will instruct the powers to be to arrange for your name and address to be removed from the mailing list. Hope you find the machine and O/S you are looking for. By the way if you are looking for an IBM compatible, the OPUS machine is good value for money, being faster more compatible, and a little cheaper than the Amstrad 1512.

Sees yas later.....Tom]

**********

Peter Booth,
34 Westville Oval,
Harrogate,
North Yourkshire,
HG1 3JW. 18/02/87

Dear Bob,

Please find enclosed a short article describing a "sideways" ROM board as used in my Interak for you to use in IUGN if you wish. I have also forwarded a copy to Dave Parkins at Greenbank for any comments he may have to make, so you may wish to wait and see if he has any comments first.

In IUGN 14, a request was made for articles and software to be submitted. I have a bit of fairly specialised software "on the go" at the moment if you think there may be an interest. At the moment I am working on a tape based personal accounts system (to be converted to disk based, CP/M compatable, as soon as I finish the transition from 8" to 3.5"), and a dedicated disk system (database?) which stores and recalls details of my record and tape collection. I also have a debugging package which mostly works and a

few other small programs. I am also working on a keyboard controller using an 8279 as I have almost completely lost C, V, and B on this keyboard.

Please could you let me know if you are interested in these sort of things as they appear (the debugger should be debugged soon!). If I promise to send you a couple of articles does that mean I might receive more than one newsletter a year?

Yours enthusiastically
Peter Booth.

P.S. I know that I've only received three newsletters since I joined in June 85, but I have not been asked for a renewal. Not that I am adverse to accepting freebies, but is a renewal due?

{Ed- I will publish your excellent article as soon as space permits. Thank you for submitting it and please accept my apologies for the late inclusion of your letter.
I am always interested in input for the newsletter. Preferably "on-disk (3.5"). My problem at the moment is a shortage of space leading to long delays before members work is published. Contact Tom about your membership, he may have lost you from his file and in time your copies will dry up if you are unable to renew.]

**********

Mr David Parkins,
C/o Greenbank Electronics,
460 New Chester Road,
Rock Ferry,
Birkenhead,
Merseyside,
L42 2AE.
Tel 051-645 3391 8th May 1987

Dear Ed,

May I through your pages reply to some of the letters and points raised in IUGN-16, and indulge in some ramblings of my own?

But first some general comments: May I say what an excellent job you are continuing to do on the production of this newsletter. As part of my official duties, I subscribe to, and read, as many competing newsletters as I can. Admittedly I am a little biased, but I must say I have never read a more interesting or well produced "amateur" publication. (I use the term "amateur" in a special sense, here to distinguish your newsletter from that of those lucky other "amateurs" who have access to fully professional phototypesetting equipment etc and a budget that can support the costs of professional printing and binding).

I hope you do not mind my revealing to the world that you produce the newsletter with only the most humble of resources: a simple Epson dot-matrix printer, "Tippex" white paint, black felt-tip pen, scissors, glue, a kitchen table and a lot of time (all our thanks must also go to Mrs Ed for her forbearance: they also serve who only stand and wait, and get the glue out of the carpet afterwards!)

I believe there is a "critical mass" which, as in a nuclear chain reaction, must be achieved before the system is able to operate with a force of its own.

To judge from the high quality of letters and comments you are receiving, to say nothing of the 12-months or more supply you already have of contributions from members, jostling for space in your pages, that "critical mass" is very close to being achieved, and an explosion of enthusiasm and interest is imminent.

(At Greenbank Electronics, where we have now had the Interak system as a product for something approaching 10 years, we have already experienced the same phenomenon - we constantly have more people requiring our goods and services than we have goods and services to provide; as fast as we get up to date in one aspect of the business, we have a backlog in the other. In our card index of customers and enquirers we have 20,000 or more names; the majority of whom are Interak enthusiasts, and we must by now have sold 10 or 20 thousands of the bare boards.)

Even after a silence of 5 years or more a customer will return to the fold, originally wooed away by the attractions of say an Ohio Superboard, a Compukit 101, Acorn Atom, Nascom 1, ETI System 68, Nascom 2, Tangerine Microtan, ETI Cortex, Wireless World SC-80, Sinclair MK-14 or ZX-80, ETI Triton, Tangerine Oric, Memotech, BBC-Model A, Flight Electronics Microprofessor, Rockwell AIM, Exidy Sorcerer, Commodore PET, Tandy TRS-80, Newbrain, Texas home computer, Dragon, Elan, Acorn Electron, Lynx, Tatung Einstein, any Japanese MX "standard" system, Practical Electronics Universal Plug in system, Advance computer, Gemini, Research Machines Z80 system, Osborne, Enterprise, Kaypro, Sharp M80Z, in fact any computer that was the best thing since sliced bread for a few heady months, but now has been de-emphasised as the manufacturer went on to better things (or bankruptcy in certain cases).

Five years ago we used to say ah yes, but will the firm (let alone the computer!) be around in 5 years' time. It takes 5 years for the penny to drop, but here we are now with more and more prodigals returning to the fold.

The IUG newsletter has obviously started a chain reaction where contributions flow in, questions are asked, answers provided, the IUGN becomes more interesting and relevant, so more people are stimulated to contribute and on it goes.

As I mentioned I am a regular reader of the newsletter (often from my privileged position being allowed to see it before it is published) but this is the first time I have felt compelled to put pen to paper (or more accurately to put wordprocessor to disk) and write in reply to some of the points made in the letters pages of IUGN 16. I hope many other members will be equally stimulated to write and comment and a whole forum can be created in these pages.

From the top then, IUGN 16.

Number 16; terrific! Who would have thought reading Issue Number 1 in December 1982 that the thing would still be in existence, getting on for 5 years later, and flourishing as it has? Why some of our very youngest members weren't even born when Interaktion began, many of our current professional project managers were

still at school or college when they first joined, and of course the Interak system itself was alive and kicking well before the group was formed.

Page 2. Notes ideas help: Just what we need. Nobody should fail to build the A.S.A.M; it shouldn't take more than half an hour or a pound or so in components (patch it onto any convenient Interak card; that's what the patch areas and bus connections are there for). Regarding the Breadboard Interface I'm a bit nervous myself about 1 metre of ribbon cable dangling on my ISBUS, but I know Bob has used that system successfully on numerous of his own experiments, including the speech synthesiser (now, tell me, why didn't anyone build that?)

Regarding the "little bird" at the top right-hand corner of page 2, I am sure that our one and only lady member would prefer not to be called a little "bird". But yes, we would love to have an article about her domestic arrangements, "Transistors over toast", "Chips with everything": if you're listening Pat(ricia), how about it?

Page 3. Where to buy CP/M software. Just what we need again. But surely there are many more suppliers with CP/M software filling their shelves crying out to be purchased before IBM takes over the world (again); doesn't anyone else know anywhere else?

For instance, Borland do some excellent CP/M software (do me a favour though, and ask for it in Interak format, don't make it easy for them and accept some other format):

eg Turbo Pascal, and numerous other "Turbo" products from Borland International (UK) Ltd, 1 Great Cumberland Place, London. W1H 7AL (Ø1)-258 3797.

Another supplier is of course yours truly, offering the Hisoft products Pascal 8Ø (Pascal compiler), and the Devpack 8Ø (Editor, Assembler, Debugger Z8Ø development suite). Ask me for Interak format and I will know what you mean.

Back issues: the charge is purely nominal (say 5Øp or so per issue), but no cheating by waiting a year and getting a year's supply for a fraction of the proper year's membership fee!

Bruce Joyce: Yet more from this prolific contributor. He is a really nice guy, not a software guru or self opinionated expert, and I'm sure he would welcome some feedback from users. Why not drop him a line? Nice to see software coming in for the disk based ZYBASIC. ZYBASIC is not direct competition for Microsoft style BASICs, but it is a fraction of the cost, and has a number of quite attractive and often unique features. Well worth a second look, even if you are a snooty type who never soils his hands with BASIC.

Page 4 Airtel session: Dialling up Bulletin Boards for the first time is a very nerve-racking procedure, so a printout of a session like this is a great comfort to those who want to do some homework first before they set foot in the lion's den. Using bulletin boards is one aspect of what computing's all about. Computing isn't just using the IBM-AT at the office to calculate your mileage expenses; computing is growing and understanding, interacting (interakting?) with other people. You know the enjoyment radio hams get out of their hobby, the same is available to you via the bulletin boards. And all the Sysops, who are invariably doing it for pleasure not reward, are delighted to receive a call from someone new. Please log on to every board in the list and tell the world you are using an Interak, let's keep it a secret no longer. (Bob and Tom have already covered the technical details of how to communicate via telephone lines in earlier issues of the newsletter, and we're all wanting to help you so don't be shy!)

Page 9 Chips Database: Another first for issue 16, a program to run under a professional package: dBase. Of course the high price of dBase means the amateur user finds it difficult to justify its purchase (Greenbank can supply dBase II to any millionaires amongst the users), but don't forget we do have a lot of professional users who must not be neglected. dBase II lets you do all the data and administrative tasks you expected the (disk based) computer to do, but couldn't manage in an inadequate language like BASIC. The trouble is, dBase II is tricky to get to grips with at first, and Frank Johnson's excellent contribution will make the first steps easy for anyone beginning (and it is a useful program in its own right). Don't forget thanks to Frank for sharing all his hard work with us entirely free of charge with no thought of reward. It is this sort of altruism amongst members which marks Interak out as truly unique.

Auntie David's Page: I cannot comment on this. Apart from the fact I have to declare an interest in what goes on here, when I saw the preliminary copy of IUGN 16 this page was blank - as usual this contributor was late sending in his material!

Page 15 Floppy Disk Transfer: Another regular contributor, Simon Waller, with more helpful information on his own unique design of floppy disk interface. Simon has correctly identified some particularly critical areas, (but note all of these problems were solved long ago in the official "Interak FDC-1" design, and operating software). Whilst acknowledging Simon's skill and enthusiasm in doing his own design, I would be very pleased to welcome him back to the fold, and if he has not already purchased the FDC-1 card I urge him to contact me personally and I shall make him an offer he cannot refuse to return to the paths of Interak righteousness. Some of the finest brains in the world (ie Bob Eldridge, David Parkins, the late Wolf Schroeder, and Kevin Daley) have conspired in their various ways to achieve mission impossible; data transfer at 5ØØ Kbits/s to and from the 2797 FDC chip by direct polling of its registers.

If the editor would like me to, I should be glad to reveal the long kept secret of how mission impossible was achieved; I think it could be very instructive (and would certainly confirm the converts faith in the hidden power of the ostensibly ugly Z8Ø instruction set. Lumpy and warty the Z8Ø's innards may be but they get you out of a jam when your back's against the wall!)

Page 16 "Doc" files: In printing these out, the Editor has performed a valuable service which is neglected by most other suppliers of public domain software: telling you what it does. There are tens of thousands of CP/M programs in the public domain, and few people have the time to investigate each one individually (looking at just one a night would take 25 years or more before they'd all been looked at, let alone run!), so it is very useful to have some preliminary selection and sorting carried out by our Editor and our Disk Librarian.

Evans Above: Another page I always enjoy reading, from an ace writer; unfortunately this too was blank in the draft copy of IUGN 16, so the contents will have to be a treat in store.

Page 23 Letter from RL Ruddock: I have some information about the Intelgraph board. It was another of those products which was launched in a fanfare of publicity, but which was gradually allowed to dwindle and die. However somebody told me that the design had been taken over by the same firm who took over the "Microtan" series of 6502/6809 cards when Tangerine finally gave up the ghost. I think it was our user Dave Ford who told me, and the address he had found for the new firm was "Mandarin Micro Systems Ltd", 48 Brightstow Road, Burnham-on-Sea, Somerset. TA8 2HP (But don't you dare ask them about the IP-68 series system, you are an Interak user remember!)

I endorse what Mr Ruddock says about Bob Eldridge's clear descriptions of CP/M versus those of Digital Research (more of these please, Bob?), but leaping to Digital Research's defence: I am sure it was Digital Research's CP/M 2.2 documentation which was being referred to. The latest CP/M, CP/M Plus (our current preferred disk operating system) has a totally new set of manuals, which although not perfect of course do give some semblance of at least being written by some inhabitant of this solar system, if not this very planet.

Scott Dadak: Evidently a second-generation Interak user. I am really delighted to hear of whole families being interested in the same computer; it is all wrong when there is just one "freak" in a family who likes computers, and the rest are all "normal". Why can't we all do it together? Thanks too to good old Ed for providing the IUGN, which allows programs such as Scott's to be published, and quite correct not to knock games as being somehow inferior examples of the programmer's arts. In fact our Ed's profound remarks on the game of life makes me think of my own favourite game: designing pcb cards, writing words, sticking pcb cards in envelopes, paying rates, answering the phone, calculating VAT, getting shouted at, getting congratulated, getting poor, getting rich, waking up, going to work, going home, going to sleep, failing to pass "go", missing a turn, retiring undefeated. I've been playing this game for years, and I'm still not fed up yet.

P Apps: What can I say? Lovely work. I try very hard myself to handle a pen legibly, and am very humbled in the presence of an expert. (By the way, it was a bit unfair of us to publish this, as it was not intended for publication, but we hope Mr Apps doesn't mind. It was worth showing to everyone, because it reminds us of the wealth of talent there is in every field in the Interak community.)

John Ritchie: Another most welcome letter from another really nice guy. (Is it owning Interak which makes people nice, or is it that nice people are drawn to Interak. Don't we have any muggers, murderers, or plain bad-tempered members?) John has let the cat out of the bag there about the station magazines, come on let's rope him in! (Don't worry about his being busy, it is a known fact that a busy man will always find time to do you a favour; someone who's not busy, by definition never does anything for anybody!)

Of course I have to tell John (and any other ZYBASIC disk users like him) that the answer to the problem of the ZYBASIC documentation is inside his own computer. As ZYBASIC on disk is in a state of "flux" (excuse the pun); currently Version 4.9, but scheduled for future changes, the manual too is in a state of flux. However it is provided on each ZYBASIC disk sent out as a file "ZYBASIC.DOC": print that out, and you've got your manual. A little bird tells me that John has no excuse to avoid the printing because he has just taken delivery of one of Greenbank's (and Epson's) finest high speed printers.

Patrick Meehan: Another constructive and helpful letter, which is extremely interesting to read. Interchange of ideas and views is what it is all about, and I think this is exactly what we're looking for. Interak is more than just a computer, it is a passport to a land of knowledge and a community of like minded enthusiasts, "all for one and one for all" etc etc. Tom's IUGN 14 contribution ("No Input = No Output") has perhaps overstated the lack of contributions and information, (in fact as our Editor has said, there is rather an embarrassment of riches in that there is already 12 months' or more supply of material sent in by members, with the flow increasing and increasing). I think Tom might have been writing as a result of being needled by one user in particular who was very keen on saying what was wrong with the group but unwilling to do anything constructive to help. Don't forget the group is run by generous hearted users in their spare time, and they are bound to feel a bit upset if the only reward they get is some bright spark saying the newsletter is not as interesting or as well produced as Personal Computer World or some such.

However Tom has performed a service in stimulating Patrick to write, I wonder would he have written if it hadn't been for Tom's pushing? Now let's have some more views from other users. It doesn't have to be technical, I myself always find the human side more interesting than the cables and disks anyway.

I myself can certainly sympathise too about the cost, although when compared to similar plug in rack-type systems Interak usually is about half the price of similar equipment. A Taiwan-made mass produced home computer with say 30 chips is bound to be cheaper than one like Interak with perhaps 200 chips. Although I have to agree that Interak even so is not an inexpensive computer, the Editor is right to point out that it saves money in the end. In fact I would go so far as to say

there is no such thing as an inexpensive computer; if you mean by "computer" what I mean by "computer". Patrick says that it has taken him a year to build 5 boards (this is by no means unusual by the way); at least he can have the satisfaction that at the end of the year he has a product which is still in production; there are very few computer products available for less than the price of Interak which do not have obsolescence inextricably built in.

Greenbank Electronics can of course supply a switch mode power supply and even a microprocessor controlled keyboard for much less than the prices found so far by Patrick. (I have sent details of the good news to him.) In fact we quite welcome the comparison with the firm "RS". This is a fair comparison. RS make millions and millions of pounds a year supplying all manner of top quality components and equipment to the professional user. The professional knows the wisdom of not trying to do things on the cheap (the professional house painter for example uses brushes which cost ten times the price the amateur is prepared to pay, and no professional wastes money) all we have to do is make sure that the right basis is used for comparison. A Sinclair C5 vehicle is cheaper than a Ford Sierra, but so what?

Ian Vaudrey Page 24: This is as our Editor says, an excellent letter, (enough to get me up and writing in reply anyway!) I am not sure whether I should rise to the bait and give a sprited defence of the Z8Ø. The reason I am a fan of the Z8Ø is simply an accident of history. When I first discovered the benefits of microprocessors in my career as a digital electronics designer, I was looking around for a development system. The only thing I found which suited me (from the point of understanding and "get-at-ability, logically as well as physically) was a system from a firm called Kemitron, the Kemitron "Universal Microprocessor Development System". In those days it had a funny old microprocessor called SC/MP (but without that SC/MP I would have learned nothing) but the CPU card could easily be changed for one with a different microprocessor. On this system I have dabbled with the 8Ø8Ø, the 68ØØ, 68Ø2, (ie near relatives of the famed 65Ø2) and of course the Z8Ø, and shortly the "super Z8Øs". The original attraction of the Kemitron system was that the bus was kept simple: address, read and write and data. No dynamic RAM was used, and no processor-specific lines, so evaluation of any particular microprocessor was very easy: unplug the old, plug in the new.

The system was such a good one that I decided to put it on general sale for the benefit of other experimenters. (Originally Kemitron had sold their bare boards through the firm Crofton Electronics, but as neither of them wanted the "bare board" business Greenbank became sole supplier.) Kemitron was a young firm then, but as they grew older and wiser they started to put the price up, and make all their cards only available as built and tested units at a couple of hundred pounds. This was quite sensible because at that sort of price they only had serious professional users, and with a good profit they could be properly supported. Kemitron Electronics is now a highly successful firm, very highly regarded, so their business plan was beyond reproach.

Greenbank Electronics on the other hand continued with the bare boards, and whilst not making much money made a good many friends (a very early customer for example was a Mr Robert Eldridge, now haven't I heard that name somewhere before?) The Z8Ø made it very easy to interface to dynamic RAM, it had a ready made ideal operating system (CP/M) and was the popular computer of the day. To this day 5Ø million Z8Ø chips per year are sold (not all by Greenbank unfortunately).

Therefore the Interak system gradually moved towards the Z8Ø exclusively, and new cards were designed by Greenbank, but without rendering the old ones obsolete (that is why for instance the VDU-K began as it did - it matched the original Kemitron VDU-A, -B, -G 3-card VDU set, which before our time was a miracle of miniaturisation compared to the the original Crofton VDU-A,-B,-C,-D,-E,-F 6-card set. Those were the days.

Kemitron Electronics was (and still is) a very practical firm, and they chose not to support interrupts on their bus, which is why the Kemitron MZB-3 does not have interrupts. (This is very practical, because interrupts are not always all they are cracked up to be; a typical software job using interrupts takes twice as long to write and perhaps 5 times as long to debug, and is more difficult to maintain and modify, and often results in a less reliable system, more prone to unexplained crashes. This is the voice of experience speaking.)

I am no Luddite however, and when you have to have interrupts you have to have them, and I can give proof that I knew of their existence and used them myself as early as 1982 (our Editor has already directed the reader to IUGN number 1 December 1982 page 8; about 5Ø pence and 5 minutes work with a soldering iron soon adds interrupts to the MZB-3 if needed).

My version of the (43-way) Kemitron bus was the 86-way ISBUS-B, details of which have been available for years (contact David Parkins if you haven't a copy of your own). This was compatible with the 43-way specification which is used on current cards (ISBUS-A), and had quite a few plans for expansion later, for example 3 interrupt lines, a 24-bit (16 Megabyte) address bus, and a 16-bit data bus. Although it would mean abandoning the Z8Ø entirely, the future has been accommodated in the form of a reallocation of the 16 data lines and 24 address lines to form a 32-bit multiplexed address and data bus, with 8 spare lines, for individual experiments with true 32-bit microprocessors.

As the editor has said, we have had to wait for some years for the CPU chip we needed to be designed. It now has been: the Zilog Z8Ø28Ø.

Also we had to wait for a disk operating system which can use the extra memory space usefully; we now have it: CP/M Plus.

The extra memory space and the CP/M Plus operating system allow further future benefits: graphics and hard disks (the graphics I am thinking of will demand hard disks). I know that talk of 16 Megabytes of RAM and hard disks will seem Cloud

Cuckoo Land to those users who are saving hard just to buy one card every two months, but they can take comfort from the fact that what they are building is a computer which has the potential to push onward to a surprisingly high standard.

There has been hitherto no point in designing a card to replace the Kemitron MZB-3 card until the time was ripe to make a significant improvement. Merely cobbling extra memory in a "paging" fashion often would hinder operation (for example in the famous BBC computer, there is a black market for the old 32K model because it actually runs most programs faster than the latest 128K model, thanks to the overhead of all that page switching; that's not for me!)

I had better get off this subject, because planning systems to last over decades is one of my dreams (not so much a dream but reality however because Interak has lasted almost a decade already and is still going more strongly than ever).

A few final shots are: the 8080 instructions aren't universally recognised as bad, in fact as our Editor says, the Zilog/Mostek mnemonics are very clear to use and remember. The strange extra instructions in the Z80 (for example one operation takes one length of time the same thing another way takes less, one instruction sets a flag, another leaves the flags the same, etc) have helped me out of a jam many times. I shall have more to say on this if I ever write the threatened article on the fast loops necessary for the 8" double density polling on the FDC-1 card. I should be most interested to see if the 6809 can do better in a tight corner like that! (One published design from Stirling Microsystems, for a 6809 in a magazine in which we used to advertise, Electronics and Computing Monthly, had to admit that they could do any combination of size and density, except the size of 8" in double density - why couldn't they do with their 6809 what we can do with a Z80A?)

And how about IBM? They are successfully selling to the world a computer using an 80386 CPU, which runs the code of a 80286, which runs the code of an 80186, an 8086 and its little brother the 8088. At the assembly language level (not the object code) the 8086 runs the same programs as the 8085 and even the 8080, so that microprocessor family wins any test of the acceptablity of the 8080 instruction set to the world at large.

A final commercial reason for our rejection of the 6809 is that hardly anyone but Ian (and one or two others) have ever asked for it, so it is impractical for us to drop the Z80 in favour of something so completely different (what would the rest of the users think?) From my own limited investigations I am prepared to agree that the 6809 is probably technically superior to the Z80, but like it or not it is the Z80 which has caught on in the 8-bit world. As far as I know there are no plans by anyone to build a "super-6809" as there are for the "super-Z80" chips, the Hitachi 64180 and 64280, the Zilog 80280, and Toshiba's multilegged super Z80.

Finally the disk library pages: Again in my preview copy of IUGN 16 there was a blank page where Charlie's Lib should be, so I can't comment on that, but I hope it will be filled by the time the newsletter is published. The idea of a continuing set of disk library update index pages in A4 loose leaf form is wonderfully simple. Other periodicals, who shall remain nameless, employ a most unworkable system of issuing a stapled and bound complete catalogue each year, so that to get the latest sheets you continually have to buy the first ones over and over again. The complete waste of paper obviously pricks their conscience so that they then save paper by putting the most incomprehensible abbreviated titles for the programs on each disk, with no explanation at all.

Our library is totally different; only updates are provided and typically two sides of an A4 sheet are used in giving a description of the contents of each disk. With this method a subscriber has a reasonable chance of determining whether or not he will be interested in the contents of the disk before ordering it. (And incidentally note the value for money; each Interak disk contains much more than the 250K or so on a typical public domain disk.)

Less paper for more information, how clever they are: Join Interaktion and save a tree.

Keep up the good work!

Yours sincerely

David M Parkins, B Eng (Hons)

[Ed - Well that is probally the longest letter I have ever published, in fact it must be close to the longest ever published in any publication. . Thank you David for an excellent commentary on the newsletter. I especially enjoyed the history of the Interak, its evolution if you like. I wonder if you could write a short piece describing the history of Greenbank in some detail. I for one would enjoy learning about how you and John started Greenbank and perhaps some of the stories about the early days of the company. Tell us, is it true that Hilda had to tell you how many sugars you take in your coffee during one absent minded moment of a major card design session?.]

**********

Bruce Joyce,
38 Castle Rd.,
Salisbury,
Wilts.
SP1 3RJ

Dear Bob

Thank you for your letter.As soon as I am 'on line' [when fundspermit] I will indeed contact you on the bulletin board.

In the meantime here are some ZYBASIC/CPM programs for the free use of the group.I don't want any money for them but I am always intrested in any software of anykind.I have a lot to learn and studying programs is a great way to do it.

Two questions.Can I easily change the printer ports on the cpm [if so how]?

Can you tell me who has the disc update for crystal basic XTAL on cpm? Crystal say that they have given it to the user group but could not tell me who.

Thankyou for the job your doing as editor. I know that this must take up a lot of time. If you still need a disc librarian I would gladly offer to help [ although maybe I should have my second drive first ].

yours faithfully

Bruce Joyce

P.S. Is it possible to have 2 * 3 1/2" drives and 1 * 5 1/4" drives on the same F.D.C. card and access them from my CP/M+?

[Ed - Sorry for printing your letter so late but I missplaced the file. I will try to answer your questions now :-

Yes the printer Ports can be changed. This is done by altering the value of the IOBYTE. You should be able to select status and data ports in Wolf's BIOS by selecting IOBYTE patterns as follows :-

| Status | Data | IOBYTE |
|---|---|---|
| 2 | 3 | ØØxxxxxx |
| Ø | 1 | Ø1xxxxxx |
| 6 | 7 | 1Øxxxxxx |
| not known | | 11xxxxxx |

If anybody knows what ports are selected for 11xxxxxx then please let us all know.

The user group does not to my knowledge have the Crystal Cp/m basic. You should contact Crystal and ask them about it.

Finally, 5.25" disks "look" the same as 3.5" disks to both Cp/m and the FDCI card and so the answer is yes the system and the FDCI will drive any mix of four drives of either diameter.]

**********

Mel Saunders,
7 Drumcliff Road,
Thurnby Lodge,
Leicester,
LE3 2LH

Date:14/5/87.

Dear Bob,

Well Bob it's been sometime since I sent you anything for Interaktion or IUGN as it now seems to be, but whatever you like to call it, it's very inportaint to the group..

I am enclosing a disk this contains some very simple Assembler and Pascal programs, and Zbasic these have been sent to Charlie Bridgstock for the disk library! I have also had a go at writing an article on Devpak8Ø Assembler/editor perhaps WE could add the odd program to the article?

Also find enclosed a full list of the data sheets/books I have available for loan to members thro' the DATABANK. Again perhaps you could use it to fill a couple of pages.

I still find I have problems with Zybasic at the moment it seems to be the DIM command; ie. Dim A(12,12), B(12), C(6) N$(5),etc what am I doing wrong? I have some programs I want to convert to Zybasic but can't do much without the Dim! See the part listing to show what I'm doing.

>LIST

```
1Ø 18764      '@@@@E OF
2Ø CLS:PAGE:DOFF
5Ø DIM M(1Ø,13): DIM N$(5)
```

I wonder if you can help some of us Xtal basic users out...

The problem is we have all have our programs on tape but now run a disk system, I have Zymon on disk but but if I load Xtal with this it just crashes, I'v tried getting Xtal onto disk but no success!!

BOB I THINK IT WOULD BE A VERY GOOD IDEA TO DATE THE ARTICLES, LETTERS, ETC THAT GET SENT INTO IUGN, AS IT IS OFTEN MANY MONTHS LATER WHEN THEY APPEAR IN PRINT, AND THEN DON'T ALWAYS SEEM AS RELATIVE!! AS THEY DID AT THE TIME.

RE DRIVE MOUNTS I HAVE NOW ARRANGED (ALONG WITH DAVID) FOR AN OLD FRIEND OF MINE, WHO IS A RETIERED ENGINEER LIVING OF COURSE IN SHEFFIELD, TO PRODUCE THESE FOR DAVID, AND SO AT LONG LAST YOU CAN MOUNT THOSE DRIVES PROPERLY!! MINE LOOK VERY GOOD, A TRUELY PROFESSIONAL JOB..contact David not me.

If you have any goodies to dump on the disk please feel free to do so.

Please note I can no longer supply the lables as per advert in IUGN No 13 well I only have stocks as per samples enclosed.

Yours Sincerely

Mel Saunders.

BITS, BOBS AND AFTERTHOUGHTS

Mel Saunders
19/May/1987

Well at long last I have now got the RTC (real time clock) working on my Interak-X. There are a few RTC chips to choose from like the MC146818 from Motorola, this chip has many good points like- 64 bytes of low power CMOS ram, an 8 bit wide data bus (many only use 4 bits) a choice of crystal frequencies, a full 1ØØ year calender, programmable interrupts and square-wave output with a frequency range of 32.768KHz to 2Hz, 12/24 hour mode.

The chip only uses the first 14 bytes of its ram so you can use the remainder for what ever you like! the first 1Ø bytes hold the current time and calender information, and bytes 1Ø-13 are for setting the clock and testing the update bit (ie. is the clock in the middle of updating the registers?, if so don't read it, it may be invalid).

Like many Motorola chips this one is designed to work on a multiplexed address-data bus (like the AY-3-8910) but programming is very straight forward ie.

```
OUT&1Ø,4:OUT&11,DATA ;   Output the hour
                         register number,
                         then data to reg
OUT&1Ø,4:H=INP(&11)  ;   Read hours reg in
                         to Var H
```

I like using 24 hour mode then I can add a AM/PM string TI$="AM"

```
IF H> 12 THEN H=H-12:TI$="PM"
```

My original idea was to add a Z8ØA CTC and clock this with the square-wave output at say 2Hz this would then provide a timing system for games and many other things!! But like all good layed plans the other half of this card is now home to Bobs

speech processor. So I will have to count the seconds some other way perhaps the clock's ram will be usefull here??

So on to something different the PSG well when I built the PSG I added a switch and a 74LS393 to switch the 4MHz clock down to 15.625KHz and many others in between. The reason for this is the AY-3-8910 should have a max frequency input of 2MHz (although it works fine at 4MHz) the lower clock frequencies will give you a lower audio range (15Hz to 61KHz) and attack/decay times of over a minute. Something a bit better would be programmable clock rates add a 74LS151 throw the switch out, use one of the I/O ports of the AY-3-8910 chip itself to select the output from the 74LS151 the eight inputs still coming from the 74LS393? (only the 3 lower bits from the sound chip port are needed)

The point of all this is to say THINK what you are doing, then you don't need to make all the corrections at a later date like ME.

Mel

[Ed- Thanks for writing Mel. Thanks also for the contributions. I will start including dates on things in the future.

Crystal do sell a version of their Basic for Cp/m. You should give them a ring to find out the details as the user group is not involved.

Your program fails on line 50 at DIM N$(5) because you have used an illegal DIM, i.e. DIM N$(5). String arrays are not supported by Zybasic. The rubbish at the front of your program is because the file has become corrupted and then saved. If you have moved this program from Crystal Basic you may find the file format differs between the two Basics. Just retype the bad lines and Zybasic will then accept them by overwriting the Crystal format with its own.]

**********

T.P.Enderson 15/3/87

Dear Mr Eldridge,

Please find enclosed an article which I hope you will consider for inclusion in the IUGN. I am a new member to the user group and in fact new to Interak itself.

I hope the information enclosed will not be too simplistic as I am sure there must be others new to Interak who could benefit from it.

If you do find my work fit for publication, I would be grateful if you could not print my address. This is for purely personal reasons, totally unconnected with Interak or the IUGN.

Yours Faithfully,

T.P. Enderson.

[Ed - Welcome to the Interak, thank you for your contribution. Your work is indeed fit for publication, please have patience and your excellent article will appear as soon as possible.]

**********

Bob Cowdery,
1 Woodland Way,
Oaklands,
Welwyn,
Herts,
AL6 0RZ. 20th Feb 1987

Dear Bob,

Thank you for your letter in reply to the "BOS" manual I sent you. I have thought how I could provide an article in machine readable form for you, but reluctantly decided it was not viable at present. I do not have any WP facilities on Interak and do not have access to any CP/M machines. It would be possible to provide DOS format disks for future articles if you are able to read these. However my ploy for the present was to keep the article as brief as possible so as to reduce your typing effort. I would like the software to be available to other users, however its not just a case of a piece of software for so much money. I think somthing along the following lines would be okay if you could arrange for this to go in the software section.

| | | |
|---|---|---|
| BOS user manual | 2.00 | 24 page manual. |
| Standard BOS binary (1 tape) | 5.00 | Assembled to CF00. VDU 2K. |
| Configuring charge | 5.00 | State require--ments. |
| Source (4-6 tapes) | 10.00 | |

Yours sincerely,

Bob Cowdery.

[Ed- If you dump the text into memory and save it with Zymon to a cassette I can read it back into CP/M. I suspect though that you are slightly behind the times as most users have moved to the CP/M operating system and would find little use for a cassette based operating system. Good luck with the program though, and please keep in touch.]

**********

M.Curtis,
192 Buxton road,
Newtown,
Nr New Mills,
Stockport,
Cheshire.

06632-4795

Dear Bob,

My reason for writing, is to leap to the defence of the 8255 P.I.O. chip. I agree with Tony Padley, a chip of this versatality and with such a long industrial record cannot be ignored.

I too have constructed an 8255 card, this has all I/O pins and power lines brought out to a 50 way D connector. This arrangement gives me maximum flexibility and allows a range of interfaces to be plugged in.

Also included here are two articles based on the 8255 card which may be of interest to other readers.

First is a ROM interface which I hope to soon upgrade to a full PROM programmer.

Second is a 4 octave music keyboard, which could be used to control the P.S.G. card.

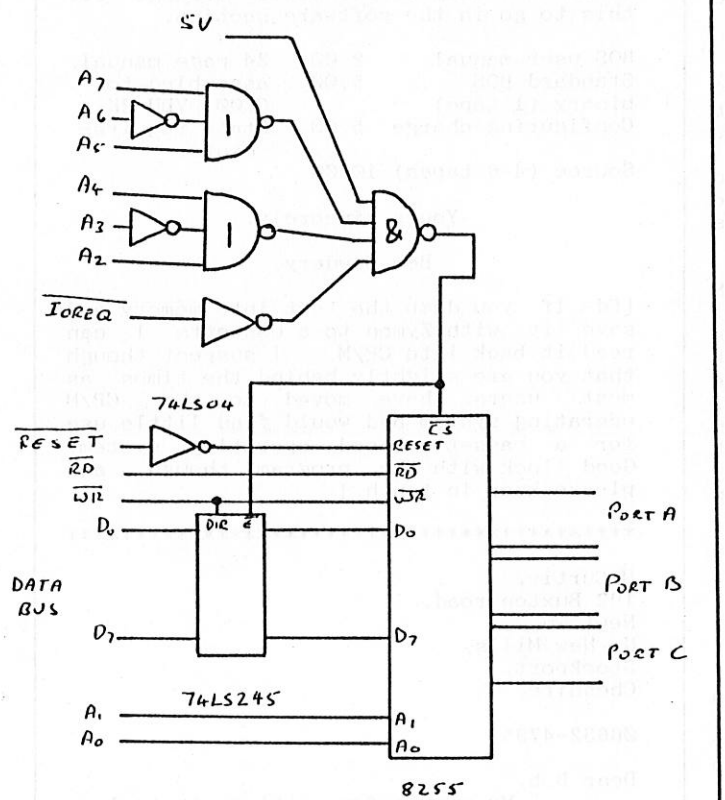
I hope to complete the programmer as soon as possible, but due to the house move my system is not yet up and running.

I hope to add disc's to my system as soon as posible and found the series of articles on this subject very interesting and informative. Though at first I was dissapointed in the choice of 3.5" drives, I personally prefer 5.25", but one look at the price of the drives has convinced me.

Looking to the future I would like to see the inclusion of a Winchester disc in the system, being far more reliable than floppies and very much larger. I have recently read some interesting application notes from Intel on this subject, using their new Winchester controller chips. I would love to develop something like this, but I doubt if my wife or bank manager would allow it.

Lastly, it seems to me that the ratio of contributers to users is very small, let's try and encorage a few more to tell us what they are doing with Interak.

Keep up the good work.
Yours Faithfully
M.Curtis




8255 BUS INTERFACE
This fixes the 8255 address at 48-4B hex. A better design would be to use a pair of 74139's making the address variable by link selection.

[Ed - Thanks, I will publish the article ASAP.
Be careful with Winchesters. How do you back them up?. What happens if a 15 megabyte drive has a crash!!. Floppies can be backed up to another floppy and a crash only looses 780k, which is probally all copied elsware.]

**********

CHARLIES LIB